

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Sistem pendataan kWh meter menggunakan pengolahan citra digital. Berikut beberapa penelitian yang terkait dengan kWh meter :

Tabel 1. Penelitian terkait

No	Nama Peneliti	Tahun	Judul	Hasil
1.	Robert Gunawan, Sri Suwarno dan Widi Hapsari	2014	Penerapan Optical Character Recognition (OCR) Untuk Pembacaan Meteran Listrik PLN	Berdasarkan hasil analisis yang dilakukan dari percobaan I dihasilkan nilai presisi sebesar 95 %. Dari percobaan II dapat disimpulkan bahwa system tidak dapat mengenali karakter yang sama apabila pola tersebut belum dilatihkan. Noise pada citra meteran memberikan pengaruh pada proses setup pola master dan pengenalan karakter terutama pada saat proses segmentasi. Dalam penerapannya, diperlukan nilai threshold yang sesuai untuk dapat melakukan setup pola master dan pengenalan karakter. Nilai threshold ini bervariasi pada tiap citra karena dipengaruhi oleh pencahayaan pada saat pengambilan citra. Semakin terang pencahayaannya maka semakin besar

				<p>pula nilai threshold yang dibutuhkan. Untuk pengambilan citra dengan jarak 15 – 20 cm dihasilkan bahwa daerah angka pemakaian pelanggan yang akan dideteksi memiliki panjang antara 110 – 120 piksel untuk meteran model lama dan 90 – 120 piksel untuk meteran model baru. Sedangkan untuk tingginya yaitu antara 10 – 15 piksel untuk meteran model lama dan 11 piksel untuk meteran model baru. Sudut pengambilan gambar juga mempengaruhi proses segmentasi karakter terutama pada citra meteran listrik model baru. Selain itu perbedaan bentuk meteran listrik model lama dan baru juga akan mempengaruhi pengenalan, untuk meteran model baru System akan mengenali sebanyak 6 karakter. Pengambilan karakter dengan menggunakan connected components labeling juga masih terdapat kekurangan dalam pengambilan karakter. Hal ini disebabkan oleh ada</p>
--	--	--	--	---

				beberapa citra meteran listrik yang karakter angkanya tidak sejajar, ada beberapa karakter yang posisinya lebih tinggi dari karakter lain.
2.	Fatsyahrina Fitriastuti dan Siswadi	2011	Aplikasi kWh (<i>Kilo What Hour</i>) Meter Berbasis Micronroller Atmega 32 Untuk Memonitor Beban Listrik	Setelah melaksanakan perancangan sistem, pembuatan dan pengamatan serta Jurnal Kompetensi Teknik Vol. 2, No. 2, Mei 2011 125 pengujian alat, dapat ditarik beberapa kesimpulan sebagai berikut : 1. Alat kWh meter digital dibangun menggunakan mikrokontroler AT Mega 32, alat ini dapat mengukur besarnya tegangan yang terjadi, besarnya arus listrik dan besar pemakaian listrik pada waktu tertentu. 2. Jalur komunikasi antara KWH Meter dengan komputer menggunakan Port serial. 3. Aplikasi ini dibangun menggunakan Borland Delphi 7 untuk merancang tampilan interface dari alat ke komputer sehingga pengguna dapat dengan mudah memantau alat kWh meter dengan MySQL untuk

				<p>menyimpan data dan ODBC (Open Database Conectivity) sebagai koneksi antara Mysql dengan perangkat lunak.</p> <p>4. KWH Meter ini dapat mengukur pemakaian daya sebanyak 4 titik yaitu 1 induk dan 3 slave.</p>
3.	Himawan Yulianto dan Teddy Marcus Zakaria	2010	Sistem Aplikasi Penghitungan kWh Meter Online	<p>1. Semua fitur – fitur yang terdapat pada sistem itu sudah dapat digunakan, yaitu :</p> <p>a. Microcontroller sudah dapat menghitung jumlah kWh dan modul GPRS sudah dapat mengirimkan jumlah KWH yang dikirim secara per hari ke server. Selain itu server juga dapat melakukan pengecekan ke dalam database.</p> <p>b. Bagian pelanggan, aplikasi berbasis web mampu melakukan fitur yang dibutuhkan diantaranya :</p> <p>1. Dapat melakukan pengecekan jumlah kWh saat ini, per hari dan juga perbulan.</p> <p>2. Dapat melakukan kegiatan forum.</p> <p>c. Bagian admin, aplikasi berbasis web mampu melakukan fitur yang dibutuhkan</p>

				<p>diantaranya:</p> <ol style="list-style-type: none"> 1. Dapat melakukan proses manajemen data pelanggan 2. Dapat melakukan proses manajemen data admin. 3. Dapat melakukan proses manajemen berita.
4.	Sukarman, M. Khoiri, Swarnada Setiawan	2009	Rancang Bangun kWh Meter digital Berbasis Mikrokontroler	<p>Dari perancangan alat ukur energi kWh meter digital berbasis mikrokontroler ATmega 8 yang dibuat dengan spesifikasi kWh meter digital sebagai berikut :</p> <ol style="list-style-type: none"> 1. Telah dilakukan rancang bangun perangkat kWh Meter Digital berbasis mikrokontroler. 2. Range beban yang terukur (0-450) watt (untuk beban diatas 450 watt belum dilakukan percobaan). 3. Ketelitian kWh meter ini ditentukan oleh tingkat kelinieritasan alat terhadap beban koreksi yaitu sebesar 0,946 dan tingkat keakuratan sebesar 0.064 dengan spesifikasi tingkat kelinieran trafo arus (sensor) sebesar 0,964 dan tingkat eror ADC internal mikrokontroler Atmega 8 sebesar 0,0001954 (0,5

				LSB). 4. Faktor kalibrasi kWh meter digital sebesar 16.
--	--	--	--	--

2.2 Landasan Teori

2.2.1 KWH Meter

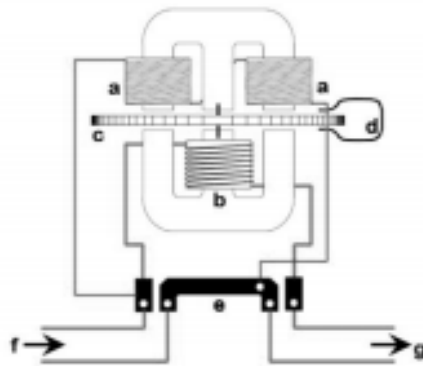
kWh meter merupakan alat ukur yang digunakan untuk menjumlah energi listrik seluruhnya yang dipakai pada waktu tertentu. Pemakaian energi listrik di industri maupun rumah tangga menggunakan satuan kWh. Alat yang digunakan untuk mengukur energi pada industri dan rumah tangga dikenal dengan *watt hour meters*. Dalam penggunaannya kWh ada dua jenis yaitu: kWh meter Pascabayar dan kWh meter Prabayar.

2.2.1.1 Kwh Meter Analog



Gambar 1. kWh Meter Analog

Gambar 1 diatas merupakan gambar dari kWh meter analog salah satu salah satu kWh meter yang biasadipakaipada tarif listrik reguler/pascabayar. Konstruksi dari kWh meter analog dapat digambarkan pada gambar 2.



Gambar 2. Konstruksi kWh meter

Bagian-bagian kWh meter dapat dijelaskan sebagai berikut:

1. Diantara piringan kWh meter ditempatkan dengan dua buah bantalan (atas dan bawah) yang berfungsi agar piringan dapat berputar dengan mendapat gesekan sekecil mungkin.
2. Rem magnet terbuat dari magnet permanen, mempunyai satu pasang kutub (utara dan selatan) yang berfungsi untuk mengerem/menetralkan ayunan perputaran piringan.
3. Roda gigi dan alat pencatat (*register*), sebagai media transmisi perputaran piringan, sehingga alat pencatat merasakan adanya perputaran untuk mencatat jumlah energi yang diukur oleh kWh meter.
4. Kumparan tegangan terdiri atas kWh meter 1 *phasa* sebanyak 1 set, KWH meter 3 *phasa* 3 kawat sebanyak 2 set dan KWH meter 3 *phasa* 4 kawat sebanyak 3 set.
5. Kumparan arus sama jumlah setnya dengan kumparan tegangan. Pada kumparan arus dilengkapi dengan kawat tahanan atau lempengan besi yang berfungsi sebagai pengatur *cosinus phi* (faktor kerja).

2.2.1.2 Kwh Meter Digital

kWh meter digital merupakan kWh meter yang dirancang dengan menggunakan komponen elektronik sebagai pemroses utama. kWh meter digital dalam penggunaannya terdapat dua jenis yaitu pascabayar dan prabayar. Cara kerja kWh meter digital pascabayar sama dengan kWh meter analog. Sedangkan kWh meter digital prabayar dilengkapi dengan *display* informasi, *keypad* untuk memasukkan angka kode *token/Stroom* atau perintah lainnya. Secara teknis operasional sistem listrik prabayar dikenal ada 2 sistem yaitu sistem 1 (satu) arah dan sistem 2 (dua) arah, perbedaan yang mendasar pada operasionalnya untuk listrik prabayar 1 (satu) arah adalah komunikasi antara meter prabayar dengan vending sistem adalah melalui media token berupa 20 digit angka yang dimasukkan pada *keypad* kWh meter prabayar, sedangkan pada sistem 2 arah komunikasi antara vending sistem dengan meter prabayar melalui media *Smart card/smart key* yang diisi ulang melalui card charger kemudian dimasukkan pada kWh meter prabayar. Salah satu contoh pada kWh meter dengan sistem 1 (satu) arah adalah KWH merek Actaris ACE9000 IBS. Berikut ini adalah fitur-fitur yang ada pada kWh meter prabayar.

Fitur standar:

1. Label Informasi : Informasi umum untuk mengetahui nomor meter, daya maksimal.
2. Indikator *LED* Rate, 1000 pulsa/kWh: Informasi untuk mengetahui ketika pulsa hampir habis,
3. Indikator *Contact* ON/OFF : Informasi untuk mengetahui status light.

4. Segel Metrologi: Informasi untuk mengetahui segel tera dan segel metrologi.
5. LCD 7 segment untuk 8 karakter : Informasi untuk pengisian Token.
6. *Keypad* dengan lapis karet.



Gambar 3. kWh meter Digital

Fitur Teknis :

1. Satu fasa 2-kawat, yaitu 1 kawat Fasa dan 1 kawat Netral,
2. *Range Voltage* : 230V 50Hz atau 120V 60Hz,
3. *Range Arus* Imin=10A dan Imax=60A,

Cara kerja kWh meter digital secara umum adalah dengan menghitung secara digital jumlah penggunaan energi listrik pelanggan. Untuk mendeteksi atau mengukur tegangan dan arus listrik digunakan sensor arus. Keluaran dari sensor tersebut akan dikonversi menjadi data digital yang kemudian akan diolah pada bagian mikrokontroler untuk menghasilkan harga atau jumlah pemakaian listrik pelanggan yang kemudian akan ditampilkan pada LCD. Selain ditampilkan pada LCD, data juga disimpan pada memori. Data yang tersimpan pada memori tidak hanya data dari kWh meter saja, tetapi juga nilai dari besaran pulsa. Besaran pulsa

didefinisikan dengan angka-angka tertentu sebagai kode voucher. Apabila kode voucher yang dimasukkan itu benar, maka besar pulsa kWh akan bertambah dan akan berkurang seiring dengan pemakaian daya PLN. Kode voucher dimasukkan melalui *keypad* dan kode yang telah dimasukkan tidak dapat digunakan lagi. Data-data ini tidak boleh hilang saat tidak ada *supply*, oleh karena itu diperlukan sebuah mikrokontroler yang memiliki EEPROM internal. *Relay* digunakan untuk memutuskan daya PLN bila pulsa prabayar habis.

2.2.2 Pengolahan Citra Digital

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar, transformasi gambar, melakukan pemilihan ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses informasi didalam citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data (T. Sutoyo, dkk, 2009).

Data atau informasi biasanya didapatkan pada suatu teks. Pada kenyataannya suatu citra dapat memberikan data atau informasi, pengolahan citra merupakan salah satu cara untuk mendapatkan data atau informasi yang di dapatkan pada suatu citra. Proses analisis data sering digunakan dalam pengambilan data pada suatu citra.

2.2.3 Citra Digital

Secara garis besar, pengolahan citra digital berlandaskan pada pemrosesan gambar dua dimensi menggunakan komputer. Citra digital adalah sebuah larik (*array*) yang berisikan atas nilai - nilai real maupun kompleks yang direpresentasikan dalam deretan bit.

Suatu citra dapat diwakili oleh fungsi $f(x,y)$ berukuran M baris dan N kolom, dimana nilai x dan y merupakan koordinat spasial. Nilai pada suatu irisan antara baris dan kolom (pada posisi x,y) disebut dengan *picture elements*, *image elements*, atau piksel. Namun, yang

lebih sering digunakan pada citra digital adalah piksel. Citra digital dapat ditulis dalam bentuk matriks sebagai berikut :

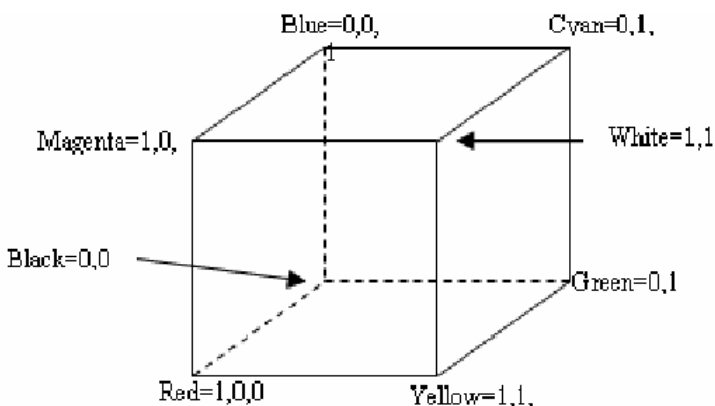
$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \vdots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix} \dots\dots\dots (1)$$

2.2.4 Citra RGB

Suatu citra RGB (Red, Green, Blue) terdiri dari tiga bidang citra yang saling lepas, masing masing terdiri dari warna utama, yaitu : merah, hijau dan biru di setiap pixel.

2.2.5 Pengolahan Warna Model RGB(*Red, Green, Blue*)

Model warna RGB berorientasi hardware, terutama untuk warna monitor dan warna pada kamera video. Dalam model warna ini tipe warna ditunjukkan dengan kombinasi tiga warna primer. Ketiga warna primer tersebut membentuk sistem koordinat kartesian tiga dimensi. Lihat Gambar 4. Subruang pada diagram tersebut menunjukkan posisi tiap warna. Nilai RGB terletak satu sudut dan nilai cyan, magenta, dan yellow berada di sudut lainnya. Warna hitam berada pada titik asal, sedang warna putih terletak pada titik terjauh dari titik asal. Grayscale membentuk garis lurus dan terletak di antara dua titik tersebut (Sutoyo, 2009).



Gambar 4. Skema warna kubik RGB

Pengolahan warna menggunakan model warna RGB sangat mudah dan sederhana, karena informasi warna dalam komputer sudah

dikemas dalam model warna yang sama. Hal yang perlu dilakukan adalah bagaimana melakukan pembacaan nilai-nilai R, G dan B pada suatu piksel, salah satu cara yang mudah untuk menghitung nilai warna dan menafsirkan hasilnya dalam model warna RGB adalah dengan melakukan normalisasi terhadap tiga komponen warna tersebut. Normalisasi penting dilakukan terutama bila sejumlah citra di ambil dengan kondisi penerangan yang berbeda. Hasil perhitungan tiap komponen warna pokok yang telah dinormalisasi akan menghilangkan pengaruh penerangan, sehingga nilai untuk setiap komponen dapat dibandingkan satu dengan lainnya walaupun berasal dari citra dengan kondisi penerangan yang berbeda, dengan catatan perbedaan tersebut tidak terlalu ekstrim. (Ahmad, 2005).

$$r(1) = \frac{R}{R+G+B} \dots\dots\dots (2)$$

$$g(2) = \frac{G}{R+G+B} \dots\dots\dots (3)$$

$$b(3) = \frac{B}{R+G+B} \dots\dots\dots (4)$$

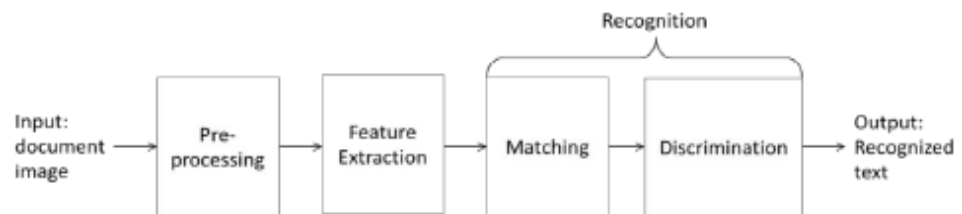
2.2.6 *Optical Character Recognition*

OCR (optical character recognition) adalah hasil dicetak atau ditulis karakter teks oleh komputer. Ini melibatkan photoscanning teks karakter demi karakter, analisis dari gambar yang dipindai dalam, dan kemudian terjemahan dari gambar karakter ke kode karakter, seperti ASCII, yang biasa digunakan dalam pengolahan data.

Dalam pengolahan OCR, gambar atau bitmap scan-in dianalisis untuk area terang dan gelap dalam rangka untuk mengidentifikasi setiap huruf abjad atau angka numerik. Ketika karakter diakui, itu diubah menjadi kode ASCII. Papan sirkuit khusus dan chip komputer yang dirancang secara tegas untuk OCR digunakan untuk mempercepat proses pengenalan.

OCR digunakan oleh perpustakaan untuk mendigitalkan dan melestarikan kepemilikan mereka. OCR juga digunakan untuk memproses cek dan slip kartu kredit dan menyortir surat. Miliaran majalah dan surat diurutkan setiap hari oleh mesin OCR, jauh mempercepat pengiriman surat.

Pada umumnya sistem OCR terdiri dari beberapa komponen yang dapat dilihat pada gambar 5.



Gambar 5. *Flowchart* Tahapan Umum Pada Sistem OCR

Teknik preprocessing diperlukan pada warna, abu-abu tingkat-atau gambar dokumen biner berisi teks dan / atau grafis. Dalam karakter sistem pengenalan sebagian besar aplikasi menggunakan gambar abu-abu atau biner sejak pengolahan gambar warna komputasi yang tinggi. Seperti itu gambar juga mengandung latar belakang dan / atau tanda air tidak seragam sehingga sulit untuk mengekstrak teks dokumen dari gambar tanpa melakukan semacam *preprocessing*, karena itu; hasil yang diinginkan dari preprocessing adalah teks gambar yang berisi biner saja.

Demikian, untuk mencapai hal ini, beberapa langkah yang diperlukan, pertama, beberapa teknik perbaikan citra untuk menghapus suara atau memperbaiki kontras pada gambar, kedua, thresholding untuk menghapus latar belakang yang mengandung adegan, watermark dan / atau kebisingan, ketiga, halaman segmentasi untuk grafis terpisah dari teks, segmentasi karakter keempat untuk karakter terpisah dari masing-masing lain dan, akhirnya, pengolahan morfologi untuk meningkatkan karakter dalam kasus dimana thresholding atau preprocessing lainnya

teknik terkikis bagian dari karakter atau ditambahkan piksel kepada mereka. Teknik di atas ini sedikit dari mereka yang dapat digunakan dalam karakter sistem pengakuan dan dalam beberapa aplikasi, beberapa atau beberapa teknik ini atau orang lain dapat digunakan pada berbagai tahap sistem OCR. Sisa bab ini akan menyajikan beberapa teknik yang digunakan selama tahap preprocessing dari sistem pengenalan karakter.

Menurut Jain, dkk., Preprocessing citra dapat melibatkan satu atau lebih dari langkah berikut (Jain et al, 2013) :

2.2.6.1 Reduksi Noise

Kemajuan teknologi yang dihasilkan perangkat akuisisi gambar dengan lebih baik perbaikan. Sementara teknologi modern telah memungkinkan untuk mengurangi tingkat kebisingan terkait dengan berbagai perangkat elektro-optik ke tingkat hampir diabaikan, masih ada beberapa sumber kebisingan yang tidak dapat dihilangkan. Gambar yang diperoleh melalui sensor yang modern mungkin terkontaminasi oleh berbagai sumber noise. Dengan suara kita lihat variasi stokastik sebagai lawan distorsi deterministik, seperti shading atau kurangnya fokus. Ada yang berbeda jenis kebisingan yang berhubungan dengan perangkat menangkap elektronik atau sumber cahaya yang digunakan jenis seperti kebisingan foton, termal, On-Chip elektronik dan kuantisasi. Sebagian besar kebisingan dapat dihilangkan oleh sensor menangkap atau kamera CCD. Sistem analisis dokumen manfaat dari pengurangan kebisingan dalam tahap preprocessing ini dapat memberikan peningkatan substansial dalam kehandalan dan ketahanan fitur tahap ekstraksi dan pengakuan dari sistem OCR. Sebuah manifestasi umum dari kebisingan digambar biner mengambil bentuk piksel terisolasi, kebisingan garam-dan-merica atau belu kebisingan, sehingga, pengolahan menghapus jenis kebisingan disebut mengisi, di mana masing-masing salt and- pixel

terisolasi pada "pulau" yang diisi oleh sekitar "laut" (O'Gorman, et al., 2008). Dalam greylevel gambar atau filter median dan low-pass filter seperti rata-rata atau Gaussian filter terbukti untuk menghilangkan noise pixel terisolasi. Gaussian blur dan filter rata-rata adalah pilihan yang lebih baik untuk memberikan tekstur halus untuk gambar. Di sisi lain, kebisingan periodik yang memanifestasikan sendiri semburan sebagai impuls-seperti yang sering terlihat di spektrum Fourier dapat disaring menggunakan notch penyaringan.

2.2.6.2 *Skew dan Perspective Correction*

Karena kemungkinan rotasi gambar input dan sensitivitas banyak dokumen metode analisis citra untuk rotasi gambar, dokumen harus diperbaiki. Teknik deteksi secara kasar dapat diklasifikasikan ke dalam kelompok berikut: analisis profil proyeksi, transformasi Hough, komponen terhubung, clustering, dan Korelasi antara teknik baris. Survei oleh Hull dan Taylor, menyelidiki dua puluh lima berbeda metode untuk deteksi gambar dokumen. Metode termasuk pendekatan berdasarkan Hough Transform analisis, profil proyeksi, distribusi titik fitur dan orientation sensitive analisis fitur. Survei itu menyimpulkan bahwa sebagian besar teknik melaporkan jarak hingga 0,1 derajat akurasi, yang membuktikan kebutuhan yang kuat untuk bekerja lebih lanjut di daerah ini untuk membantu menunjukkan kekuatan dan kelemahan dari algoritma individu (Hull & Taylor, 1998). Di Selain itu, ada teknik baru yang muncul untuk aplikasi tertentu seperti metode Al-Shatnawi dan Omar yang didasarkan pada pusat gravitasi untuk menangani Arabgambar dokumen (Al-Shatnawi & Omar, 2009). Oleh karena itu, pilihan menggunakan condong sebuah teknik deteksi / koreksi tergantung pada aplikasi dan jenis gambar yang digunakan.

2.2.6.3 Binerisasi

Sebuah proses konversi citra berwarna atau citra *grayscale* ke suatu citra bi-level. Setiap piksel dikategorikan sebagai salah satu latar depan atau latar belakang (Sezgin, 2004).

2.2.7 Citra Biner

Sebuah citra biner adalah sebuah gambar di mana setiap pixel mengasumsikan salah satu dari hanya dua nilai diskrit. Pada dasarnya, kedua nilai sesuai dengan dan mematikan. Melihat gambar dengan cara ini membuat lebih mudah untuk membedakan fitur struktural. Sebagai contoh, dalam gambar biner, mudah untuk membedakan objek dari latar belakang.

Dalam Image Processing Toolbox, citra biner disimpan sebagai matriks dua dimensi 0 (yang mewakili off piksel) dan 1 (yang mewakili pada piksel). Pada piksel latar depan gambar, dan off piksel latar belakang.

Operasi citra biner kembali informasi tentang bentuk atau struktur gambar biner saja. Untuk melakukan operasi ini pada jenis lain dari gambar, Anda harus terlebih dahulu dikonversi ke biner (menggunakan, misalnya, fungsi `im2bw`).

2.2.8 Citra *Grayscale*

Grayscale adalah berbagai nuansa abu-abu tanpa warna jelas. Mungkin warna paling gelap hitam, yang merupakan total tidak adanya cahaya yang ditransmisikan atau dipantulkan. Mungkin naungan ringan putih, transmisi keseluruhan atau pantulan cahaya sama sekali terlihat panjang gelombang s . Nuansa abu-abu menengah diwakili oleh tingkat yang sama kecerahan tiga warna primer (merah, hijau dan biru) untuk cahaya yang ditransmisikan, atau jumlah yang sama dari tiga pigmen utama (cyan, magenta dan kuning) untuk cahaya yang dipantulkan. Untuk melakukan perubahan suatu gambar full color (RGB) menjadi suatu citra *grayscale* (gambar keabuan).

2.2.9 Metode Otsu

Metode Otsu merupakan salah satu metode untuk segmentasi citra digital dengan menggunakan nilai ambang secara otomatis, yakni mengubah citra digital warna abu-abu menjadi hitam putih berdasarkan perbandingan nilai ambang dengan nilai warna piksel citra digital. Metode Otsu *thresholding* diperkenalkan pertama kali oleh Nobuyuki Otsu, dalam jurnal ilmiahnya yang berjudul “*A Threshold Selection Method from Grayscale Histogram*” pada tahun 1979.

Untuk mendapatkan nilai *threshold* ada perhitungan yang harus dilakukan. Langkah awal yang harus dilakukan adalah membuat histogram. Dari histogram dapat diketahui jumlah piksel untuk setiap tingkat keabuan. Tingkat keabuan citra dinyatakan dengan i sampai dengan L . Level ke i dimulai dari 1, yaitu piksel 0. Untuk L , maksimal level adalah 256 dengan piksel bernilai 255. Nilai ambang yang akan dicari dari suatu citra *grayscale* dinyatakan dengan k . Nilai k berkisar antara 0 sampai dengan $L-1$, dengan nilai $L=256$ (simbol histogram adalah P_i). Jadi probabilitas setiap piksel pada level ke i dinyatakan dengan persamaan:

$$P_i = \frac{n_i}{N} \dots\dots\dots(5)$$

Keterangan:

P_i = Probabilitas piksel ke- i

n_i = Jumlah piksel dengan tingkat keabuan i

N = Total jumlah piksel pada citra

2.2.10 Thresholding

Thresholding adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk obyek dan *background* dari citra secara jelas. Citra hasil *thresholding* biasanya digunakan lebih lanjut untuk proses pengenalan obyek serta ekstraksi fitur. Metode *thresholding* secara umum dibagi menjadi dua, yaitu :

2.2.10.1 Thresholding Global

Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah *threshold* (batas ambang) global T , yang berlaku untuk seluruh bagian pada citra.

2.2.10.2 Thresholding Adaptif

Thesholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan *threshold* yang berbeda.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \dots\dots\dots(6)$$

2.2.11 Normalisasi

Pengaturan kontras dan intensitas cahaya atau normalisasi dilakukan dengan mengurangi perbedaan kekuatan penerangan dan dampak dari derau (*noise*) pada sensor. Metode operasi *pixel* berikut dapat digunakan untuk mengatasi hal tersebut.

$$\Gamma(x,y) = \begin{cases} \phi_d + \lambda & \text{Jika } I(x,y) > \phi \\ \phi_d - \lambda & \text{sebaliknya} \end{cases} \dots\dots\dots(7)$$

Dengan

$$\lambda = \sqrt{\frac{\rho_d \{I(x,y) - \phi\}^2}{\rho}} \dots\dots\dots(8)$$

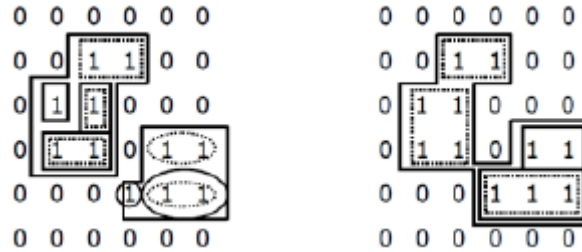
Dengan Γ adalah citra hasil, I adalah citra asal, ρ dan adalah rata-rata dan varians dari citra asal, serta dan adalah rata-rata dan varians citra hasil yang diinginkan (diinputkan).

2.2.12 Segmentasi Citra

Tujuan dari segmentasi citra adalah untuk cluster piksel ke daerah gambar yang menonjol, yaitu, daerah sesuai dengan permukaan individu, benda, atau bagian alami dari objek.

Segmentasi yang dapat digunakan untuk pengenalan obyek, batas oklusi estimasi dalam gerakan atau sistem stereo, kompresi gambar, editing gambar, atau database gambar look-up.

Segmentasi didasarkan pada pengukuran yang diambil dari gambar dan mungkin abu-abu tingkat, warna, tekstur, kedalaman atau gerak.



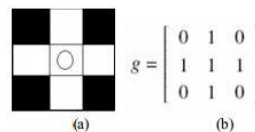
Gambar 6. Koordinat Citra Digital

2.2.13 Morfologi Citra

Kata dari *morphology* umumnya menunjukkan cabang dari ilmu biologi yang berhubungan dengan bentuk dan struktur tumbuhan dan hewan. Didalam pengolahan citra istilah dari *mathematic morphology* digunakan sebagai *tools* untuk mengekstraksi komponen citra yang berguna untuk deskripsi dan representasi dari region shape seperti *boundaries*, *skeletons* dan *convex hull*. Proses *morphology* juga berguna untuk *pre* atau *post-processing* seperti *thinning*, *filtering*, *pruning*.

2.2.13.1 Structuring Element

Structuring element (strel) dapat diibaratkan dengan mask pada pemrosesan citra biasa (bukan secara morfologi). Strel juga memiliki titik poros (disebut juga titik origin). Titik origin ditandai dengan tanda titik hitam. Jika tidak ada tanda titik hitam maka diasumsikan titik origin berada di pusat simetri.



Gambar 7. Strel (a) titik “O” adalah titik poros, (b) representasi biner strel

2.2.13.2 Operasi-Operasi Morfologi

Dalam morfologi ada beberapa operasi yang dapat dilakukan, yaitu :

1. Dilasi

Dilasi didefinisikan sebagai proses “penumbuhan” atau “penebalan” objek citra biner. Misalkan A dan B adalah himpunan-himpunan piksel. Dilasi A oleh B dinotasikan dengan $A \oplus B$ dan didefinisikan dengan:

$$A \oplus B = \bigcup_{(u,v) \in B} A(u, v) \dots\dots\dots(9)$$

Ini berarti bahwa untuk setiap titik pada A dilakukan translasi atau pergeseran pada arah (u, v) dan kemudian menggabungkan seluruh hasil pergeseran (union)

2. Erosi

Erosi merupakan proses mengecilkan atau menipiskan objek citra biner. Misalkan A dan B adalah himpunan-himpunan piksel. Erosi A oleh B dinotasikan dengan $A \ominus B$ dan didefinisikan dengan :

$$A \ominus B = \{w: B_w \subseteq A\} \dots\dots\dots(10)$$

3. Closing

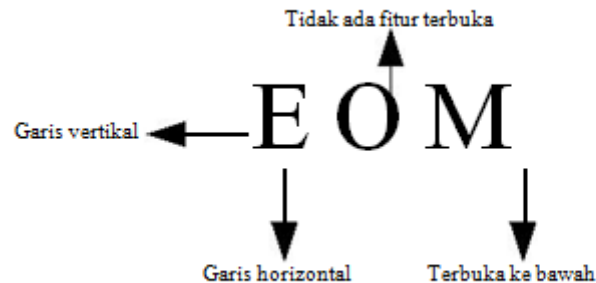
Operasi closing adalah kombinasi antara operasi dilasi dan erosi yang dilakukan secara berurutan. Citra asli didilasi terlebih dahulu, kemudian hasilnya dierosi. Proses closing pada sebuah citra A oleh strel B dinotasikan dengan $A \bullet B$ dan didefinisikan sebagai :

$$A \bullet B = (A \oplus B) \ominus B \dots\dots\dots(11)$$

2.2.14 Ekstraksi Fitur

Feature extraction atau ekstraksi fitur merupakan salah satu cara untuk mengenali suatu objek dengan melihat ciri-ciri khusus yang dimiliki objek tersebut. Tujuan dari *feature extraction* adalah melakukan dan

perbandingan yang bisa digunakan untuk mengklasifikasikan ciri-ciri yang dimiliki oleh suatu citra.



Gambar 8. Ilustrasi feature extraction

Gambar 8 merupakan ilustrasi citra karakter dengan ciri-cirinya. Ciri-ciri dari masing-masing citra *template* akan di simpan. Citra masukan yang akan dibandingkan akan dianalisis berdasarkan ciri-ciri citra. Ciri-ciri yang dimiliki citra masukan akan diklasifikasikan terhadap ciri-ciri citra *template*.

2.2.15 *Template Matching*

Template matching adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan). Teknik ini banyak digunakan dalam bidang industri sebagai bagian dari *quality control*.

Metode *template matching* adalah salah satu metode terapan dari teknik konvolusi. Metode ini sering digunakan untuk mengidentifikasi citra karakter huruf, angka, sidik jari (*fingerprint*) dan aplikasi-aplikasi pencocokan citra lainnya. Secara umum teknik konvolusi didefinisikan sebagai suatu cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka ke tiga.

Teknik konvolusi pada metode *template matching* dilakukan dengan mengkombinasikan deret angka dari citra masukan PCB yang berupa aras keabuan dengan citra PCB sumber acuan berupa aras keabuan, hingga akan didapatkan nilai korelasi yang besarnya antara -1 dan +1.

Penerapan metode *template matching* pada identifikasi kecacatan PCB dapat dilakukan dengan langkah utama sbb:

1. Pengepasan posisi: Dilakukan dengan mencuplik 80% area citra untuk mendapatkan posisi ideal.
2. Hitung nilai korelasi silang : Untuk mengklasifikasikan suatu citra PCB adalah baik dan tanpa cacat sedikitpun, maka nilai korelasi adalah 1 dan cacat total maka nilai korelasinya adalah -1. Rumus yang digunakan adalah :

$$r = \frac{\sum_{i=0}^{N-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{N-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{N-1} (y_i - \bar{y})^2}} \dots\dots\dots(12)$$

Dengan :

- x = *Template* berupa citra keabuan
- \bar{x} = Nilai rata-rata *template* berupa citra keabuan
- y = Citra sumber berupa citra keabuan
- \bar{y} = Nilai rata-rata citra sumber, citra keabuan
- N = Jumlah pixel pada citra
- r = Nilai korelasi silang pada citra aras keabuan

3. Deteksi akhir : Dari nilai korelasi yang didapat, nilai tersebut kemudian di konversikan dalam rentang 0 sampai 255 pada channel red untuk digambarkan dalam bentuk segiempat pada titik koordinat citra PCB yang mengalami cacat.

Prinsip metode ini adalah membandingkan antara image objek yang akan dikenali dengan image *template* yang ada. Image objek yang akan dikenali mempunyai tingkat kemiripan sendiri terhadap masing-masing image *template*. Pengenalan dilakukan dengan melihat nilai tingkat kemiripan tertinggi dan nilai batas ambang pengenalan dari image objek tersebut. Bila nilai tingkat kemiripan berada di bawah nilai

batas ambang maka image objek tersebut dikategorikan sebagai objek tidak dikenal.

Selanjutnya untuk dapat mengimplementasikan metode *template matching* maka perlu dilakukan sejumlah operasi pengolahan citra digital, antara lain:

Penapisan Citra (Filtering) : dilakukan bila citra yang akan dianalisis memiliki derau sehingga perlu dihaluskan dengan tapis citra. Perancangan tapis dengan memanipulasi piksel-piksel tetangga membuat citra lebih halus, bentuk sudut, dan tepi citra tetap terjaga. Pada proses perekaman citra digital dapat terjadi gangguan yang bersifat frekuensi rendah, dimana terjadi proses pemerataan intensitas cahaya pada suatu titik sampel dengan titik-titik tetangganya. Gangguan lain yang sering terjadi pada proses perekaman citra digital adalah terjadinya gangguan berbentuk garis-garis akibat adanya kerusakan pada sebagian detektor sensor. Juga sering dijumpai gangguan lain dalam bentuk bercak hitam yang acak. Pengambangan (Tresholding) : digunakan untuk mengubah citra dengan format keabuan yang mempunyai nilai lebih dari dua ke format citra biner yang hanya memiliki dua nilai (0 atau 1). Dalam hal ini titik dengan rentang nilai keabuan tertentu diubah menjadi warna hitam dan sisanya menjadi warna putih atau sebaliknya.

2.2.16 Klasifikasi

Jika tujuan dari feature extraction adalah memetakan pola berdasarkan ciri-ciri yang dimiliki oleh suatu citra, maka klasifikasi bertujuan untuk mengenali citra dengan cara mengklasifikasikan ciri-ciri yang dimilikinya.

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan

untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.

Proses klasifikasi biasanya dibagi menjadi dua fase yaitu fase learning dan fase test. Pada fase learning, sebagian data yang telah diketahui kelas datanya diumpangkan untuk membentuk model perkiraan. Kemudian pada fase test model yang sudah terbentuk diuji dengan sebagian data lainnya untuk mengetahui akurasi dari model tersebut. Bila akurasinya mencukupi model ini dapat dipakai untuk prediksi kelas data yang belum diketahui.

Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk menghitung jarak antara ciri-ciri citra template dan citra masukan.

2.2.17 Tesseract

Tesseract mungkin yang paling akurat mesin OCR open source yang tersedia . Dikombinasikan dengan Leptonica Image Processing Library dapat membaca berbagai format gambar dan mengkonversikannya ke teks di lebih dari 60 bahasa . Itu salah satu dari 3 mesin di tes Akurasi 1995 UNLV . Antara tahun 1995 dan 2006 telah sedikit pekerjaan yang dilakukan di atasnya, tapi sejak saat itu telah diperbaiki secara luas oleh Google.

2.2.16.1 Arsitektur Tesseract

Sejak HP memiliki independen maju tata letak halaman teknologi analisis yang digunakan dalam produk, (dankarena itu tidak dirilis untuk open-source) Tesseract pernahdiperlukan analisis sendiri tata letak halaman. Tesseract Oleh karena itu mengasumsikan bahwa input adalah citra biner denganopsional daerah teks poligonal didefinisikan.

Pengolahan berikut langkah-demi-langkah, tetapi beberapa tahapan sudah berbeda. Langkah pertama adalah analisis komponen

terhubung di mana garis dari komponen disimpan. Ini adalah komputasi Keputusan desain, tetapi memiliki keuntungan yang signifikan dengan inspeksi dari peguraian adalah sederhana untuk mendeteksi teks terbalik dan mengenalinya dengan mudah seperti teks hitam-putih. Tesseract mungkin mesin OCR pertama mampu menangani putih-on-hitam teks sehingga sepele. Pada tahap ini, menguraikan kumpulan, murni, menjadi gumpalan.

Gumpalan tersebut akan disusun dalam baris teks, dan garis-garis dan daerah dianalisis untuk lapangan tetap atau proporsional teks. Baris teks yang dipecah menjadi kata-kata yang berbedamenurut jenis spasi karakter. Teks dipecah menjadi kata-kata yang pasti menggunakan ruang dan ruang fuzzy. Pengujian kemudian mulai sebagai proses dua-pass.

Dilakukan usaha untuk mengenali setiap kata. Setiap kata yang dilewatkan kepengklasifikasi adaptif sebagai data pelatihan. Adaptif classifier kemudian mendapat kesempatan untuk lebih akurat mengenali teks lebih rendah ke bawah halaman. Karena classifier adaptif berguna memperlambat untuk memberikan kontribusi dekat bagian atas halaman, kedua lulus dijalankan di atas halaman, di mana kata-kata yang tidak diakui cukup baik diakui lagi.

Sebuah tahap akhir menyelesaikan ruang fuzzy, dan pemeriksaan hipotesis alternatif untuk x-height untuk menemukan SmallCap teks.

2.2.16.2 Pencarian Teks-Line dan Kata

Algoritma line finding dirancang supaya halaman yang miring dapat dikenali tanpa harus de-skew (proses untuk mengubah halaman yang miring menjadi tegak lurus) sehingga tidak menurunkan kualitas gambar. Kunci bagian proses ini adalah blob filtering dan line construction. (Smith, 2009,p1).

Filtered blob lebih cenderung cocok dengan model non-overlapping, parallel, tetapi berupa garis-garis miring (sloping line). Pemrosesan blob oleh koordinat x memungkinkan untuk menetapkan blob ke sebuah baris teks yang unik. Sementara penelusuran kemiringan di seluruh halaman, dengan banyak mengurangi bahaya penugasan ke baris teks yang salah dengan adanya kemiringan (skew). Setelah blob tersaring ditetapkan ke garis, sebuah median terkecil dari kotak-kotak yang cocok digunakan untuk memperkirakan baseline, dan blob yang sudah difilter dengan baik dipasang kembali ke garis yang sesuai. (Ray Smith, 2009, p2).

Langkah terakhir dari proses pembuatan garis (line creation) adalah menggabungkan blob yang overlapping, menempatkan diacritical marks dengan dasar yang tepat, dan menghubungkan bagian-bagian dari beberapa karakter yang rusak secara benar. (Ray Smith, 2009, p2).

2.2.16.3 *Baseline Fitting*

Setelah baris teks telah ditemukan, garis pangkal (*baseline*) dicocokkan secara lebih tepat menggunakan *quadratic spline*. Hal ini merupakan salah satu kelebihan sistem OCR dan memungkinkan *tesseract* untuk menangani halaman dengan garis pangkal (*baseline*) yang miring. (Ray Smith, 2009,p2)

Baseline dicocokkan oleh partisi *blob* menjadi beberapa kelompok dengan sebuah perpindahan kontinu yang cukup layak untuk garis pangkal lurus yang asli. *Quadratic spline* dicocokkan ke partisi yang paling padat (diasumsikan sebagai *baseline*) dengan kuadrat terkecil. *Quadratic spline* memiliki keuntungan bahwa perhitungan ini cukup stabil tetapi merugikan jika muncul diskontinuitas ketika beberapa segmen *spline* diperlukan. Dalam hal ini, *cubic spline* bekerja lebih baik. (Ray Smith, 2009,p2)

Volume 69, pages 872-879.

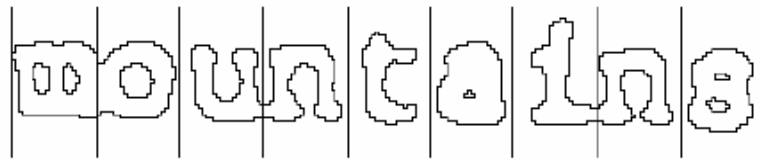
Gambar 9. Halaman Dengan Baseline Miring

2.2.16.4 Perkiraan Ketinggian X Pada Teks

Setelah menemukan baris teks dan menyusun blok blob menjadi baris-baris, Tesseract mengestimasi ketinggian-x untuk setiap baris teks. Pertama, algoritma estimasi ketinggian-x menentukan batas-batas maksimum dan minimum dari ketinggian-x yang dapat diterima berdasarkan ukuran garis inisial yang dihitung untuk blok. Kemudian, setiap baris secara terpisah, ketinggian bounding box blob terjadi pada garis dikuantisasi dan dikumpulkan menjadi sebuah histogram. Dari histogram ini, algoritma pencarian ketinggian-x mencari ketinggian dua mode yang paling sering terjadi yang cukup jauh terpisah untuk menjadi ketinggian-x dan ketinggian-ascender. Untuk mengantisipasi noise, algoritma memastikan mode ketinggian yang diambil menjadi ketinggian-x dan ketinggian-ascender memiliki jumlah yang cukup atau kejadian- kejadian relatif terhadap jumlah keseluruhan blob pada baris.

2.2.16.5 Chopping atau Pematongan Karakter

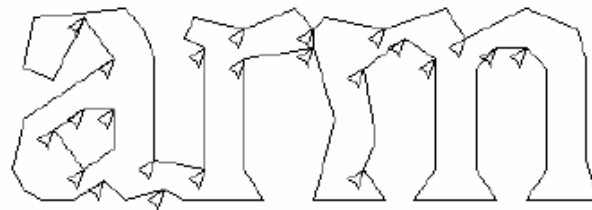
Tesseract menguji garis teks (*text line*) untuk menentukan apakah mereka merupakan *fixed pitch*. Bila ditemukan *fixed pitch text*, *tesseract* memotong kata-kata menjadi karakter-karakter. (Ray Smith, 2009,p2)



Gambar 10. Pemotongan karakter

2.2.16.6 Pemisahan Karakter Terhubung

Apabila hasil dari pengenalan kata tidak memuaskan, *tesseract* berusaha untuk memperbaiki hasil dengan memisahkan *blob* dengan keyakinan terburuk dari pengklasifikasian (*classifier*) karakter. Kandidat untuk titik-titik pemisahan ditemukan dari simpul cekung dari pendekatan poligonal *outline* dan mungkin saja terdapat titik cekung berlawanan lainnya atau segmen garis. Ini akan menghabiskan sampai 3 pasang titik pemotongan untuk memisahkan karakter yang terhubung dari *set ASCII*. (Ray Smith, 2009,p3)



Gambar 11. Kandidat Titik Potong

Gambar 2.6 di atas menunjukkan satu set calon titik potong (*chop points*) dengan tanda panah dan potongan terpilih sebagai sebuah garis melintasi kerangka dimana huruf 'r' bersentuhan dengan 'm'. (Ray Smith, 2005,p3)

2.2.16.7 Asosiasi Karakter Patah

Ketika potongan yang potensial tidak ada lagi, ketika kata tersebut masih belum cukup baik, hal ini diberikan kepada *associator*. *Associator* membuat pencarian A^* (*best first search*)

dari segmentasi grafik yang mungkin kombinasi dari *blob* yang dipotong secara maksimal ke dalam kandidat karakter. Ketika A* segmentation digunakan untuk diimplementasikan pertama kali pada tahun 1989, akurasi *tesseract* terhadap karakter yang rusak cukup baik yang menjadikan *tesseract* mesin komersial pada saat itu. (Ray Smith, 2005,p3)



Gambar 12. Sebuah kata yang rusak bisa dikenali

2.2.16.8 Klasifikasi Bentuk

a. *Static Classifier*

Sebuah versi awal dari *tesseract* digunakan topologi fitur yang dikembangkan dari karya Shillman. Ide selanjutnya melibatkan penggunaan segmen dari polygonal pendekatan sebagai fitur, tapi pendekatan ini juga tidak cukup kuat untuk karakter yang rusak. Solusi terobosan yang digunakan adalah gagasan bahwa fitur yang tidak diketahui tidak perlu sama dengan fitur dalam data pelatihan.

Selama pelatihan, segmen dari pendekatan poligonal digunakan untuk feature, namun pada proses pengenalan, feature kecil yang panjangnya tetap (dalam unit ternormalisasi) diekstraksi dari outline dan dicocokkan secara many-to-one terhadap prototipe feature yang ter-cluster pada data pelatihan. (Ray Smith, 2009,p3)

b. *Adaptive Classifier*

Tesseract tidak menggunakan template classifier, tetapi menggunakan feature yang sama seperti static classifier. Perbedaan yang signifikan antara static classifier dan adaptive

classifier, terlepas dari data pelatihan, adaptive classifier menggunakan normalisasi isotropic baseline/x-height, sedangkan static classifier menormalisasi karakter oleh centroid (momen pertama) untuk posisi dan momen kedua untuk normalisasi ukuran yang anisotropic. (Ray Smith, 2005, p4)

Feature merupakan komponen pendekatan poligonal dari outline sebuah bentuk. Pada training, vektor fitur 4 dimensi (x, posisi-y, arah, panjang) diturunkan dari setiap elemen pendekatan poligonal dan dikelompokkan untuk membentuk prototipikal vektor fitur. Pada pengenalan, elemen-elemen poligon dipecah menjadi bagian-bagian yang lebih pendek dengan panjang yang sama, sehingga dimensi panjang dieliminasi dari vektor fitur. Beberapa fitur pendek dicocokkan dengan setiap fitur prototipikal dari training, hal ini membuat proses klasifikasi lebih kuat terhadap karakter yang terputus. (Ray Smith, 2005, p5)



Gambar 13. Prototipe huruf 'h'

2.2.18 Pengukuran Kinerja Klasifikasi

Dalam pemrosesan sinyal, korelasi silang adalah ukuran kesamaan dari dua seri sebagai fungsi dari lag satu relatif terhadap yang lain. Hal ini juga dikenal sebagai titik produk geser atau sliding batin - produk. Hal ini biasanya digunakan untuk mencari sinyal panjang untuk, fitur lebih pendek dikenal. Ini memiliki aplikasi dalam pengenalan pola, analisis partikel tunggal, tomografi elektron, rata-rata, pembacaan sandi dan neurofisiologi.

Untuk fungsi kontinu f dan g , korelasi silang didefinisikan sebagai :

$$(f \star g)(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f^*(t) g(t + \tau) dt, \dots\dots\dots(13)$$

Dimana f^* menunjukkan konjugasi kompleks f dan T adalah lag.

Demikian pula, untuk fungsi diskrit, korelasi silang didefinisikan sebagai :

$$(f \star g)[n] \stackrel{\text{def}}{=} \sum_{m=-\infty}^{\infty} f^*[m] g[m + n]. \dots\dots\dots(14)$$

Korelasi silang mirip di alam untuk konvolusi dari dua fungsi. Dalam autokorelasi, yang merupakan korelasi silang dari sinyal dengan dirinya sendiri, akan selalu ada puncak pada lag nol, dan ukurannya akan menjadi kekuatan sinyal.

Probabilitas dan statistik, istilah cross-korelasi digunakan untuk mengacu pada korelasi antara entri dari dua random vektor X dan Y , sedangkan autocorrelations dari vektor acak X dianggap korelasi antara entri dari X sendiri, mereka membentuk matriks korelasi (matriks korelasi) dari X . Hal ini analog dengan perbedaan antara autokovarian dari vektor acak dan cross-kovarians dari dua vektor acak. Satu perbedaan yang lebih untuk menunjukkan bahwa probabilitas dan statistik definisi korelasi selalu mencakup faktor standardisasi sedemikian rupa bahwa korelasi memiliki nilai antara -1 dan +1.

Jika X dan Y adalah dua variabel acak independen dengan fungsi densitas probabilitas f dan g , masing-masing, maka kepadatan probabilitas perbedaan $Y - X$ secara resmi diberikan oleh korelasi silang (dalam arti sinyal-pengolahan) $f \star g$; Namun terminologi ini tidak digunakan dalam probabilitas dan statistik. Sebaliknya, lilit $f \star g$ (setara dengan korelasi silang dari $f(t)$ dan $g(t)$) memberikan fungsi kepadatan probabilitas jumlah $X + Y$.

2.2.19 Black-Box(*Testing*)

Tester menggunakan behavioral test (disebut juga Black-Box Tests), sering digunakan untuk menemukan bug dalam high level operations, pada tingkatan fitur, profil operasional dan skenario customer. Tester dapat membuat pengujian fungsional black box berdasarkan pada apa yang harus sistem lakukan. Behavioral testing melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. Behavioral test paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan bug umum untuk jenis desain.

Black box testing juga disebut functional testing, sebuah teknik pengujian fungsional yang merancang test case berdasarkan informasi dari spesifikasi.

2.2.20 Xampp

Xampp adalah paket gratis layanan web yang dikembangkan oleh Apache. Paket tersebut cross-platform, sehingga dapat bekerja pada Windows, Mac OS X, Solaris dan Linux tentu saja. Tujuan utama dari paket ini adalah untuk menyederhanakan instalasi dan kontrol beberapa layanan web.

Yang termasuk Apache HTTPD (2.2.1.1), OpenSSL 0.9.8i, MySQL 5.1.33, PHP 5.2.9. Pada awalnya dirancang sebagai aplikasi pengembangan, sehingga orang bisa menguji skrip, kode dan website pada komputer mereka sendiri tanpa perlu server eksternal menggunakan semua layanan yang diperlukan. Aplikasi ini sangat mudah untuk menginstal, hanya menjalankan installer akan memungkinkan untuk memilih layanan yang ingin dijalankan di komputer, dan ketika untuk menjalankannya. Setelah program ini terinstal, dapat menggunakan aplikasi Control Panel untuk memulai, menghentikan atau mengelola layanan dengan mudah. Ini

fitur antarmuka grafis yang sangat berguna, tetapi ada juga perintah terminal kuat yang dapat Anda gunakan untuk mengelola semua layanan.

2.2.21 Android

Android adalah sistem operasi berbasis linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian dibelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008 (Wikipedia Android, 2014).

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survey pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi seluler (Wikipedia Android, 2014).

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Akibatnya, meskipun pada awalnya sistem operasi ini dirancang khusus untuk telepon pintar dan tablet, Android juga dikembangkan menjadi aplikasi tambahan di televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Sifat Android yang terbuka telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain (Wikipedia Android, 2014).



Gambar 14. Logo Android

2.2.21.1 Versi Android

Android memiliki sejumlah pembaharuan semenjak rilis aslinya. Pembaharuan inidilakukan untuk memperbaiki bug dan menambah fitur-fitur yang baru. Berikut merupakan versi-versi yang dimiliki Android sampai saat ini:

Tabel 2.Versi Android

Nama Versi	Versi	Tahun Rilis
Alpha	1.0 (Pre-Commercial)	2007-2008
Beta	1.0 (Pre-Commercial)	2007-2008
Cupcake	1.5	2009
Donut	1.6	2009
Éclair	2.0-2.1	2009-2010
Froyo	2.2-2.2.3	2010-2011
Gingerbread	2.3-2.3.7	2010-2011
Honeycomb	3.0-3.2.6	2011-2012
Ice Cream Sandwich	4.0-4.0.4	2011-2012
Jelly Bean	4.1-4.3.1	2013
KitKat	4.4-4.4.4	2013
Lollipop	5.0	2014