

# APLIKASI MOBILE PENGENALAN CITRA MENGUNAKAN METODE *LEARNING VECTOR QUANTIZATION*

Irawan Afrianto<sup>1)</sup>, Devi Priatama<sup>2)</sup>

<sup>1,2)</sup> Teknik Informatika Universitas Komputer Indonesia (UNIKOM) Bandung  
Jl. Dipati Ukur 112-116 Bandung 40132  
email : irawan\_afrianto@yahoo.com<sup>1)</sup>, devy.priatama@gmail.com<sup>2)</sup>

## Abstrak

Kebutuhan akan informasi yang cepat dan dapat diakses dimana dapat dilakukan dengan membangun aplikasi berbasis mobile. Salah satu kebutuhan adalah mencari informasi-informasi yang terdapat di internet dengan bantuan aplikasi mobile.

Teknologi pengenalan citra biasanya berbentuk aplikasi desktop yang digunakan di lingkungan yang terbatas. Dengan teknologi mobile, sistem pengenalan citra dapat dikembangkan dalam bentuk aplikasi mobile dengan maksud dapat diakses lebih dinamis, kapan saja dan dimana saja. Dalam melakukan pengenalan citra dibutuhkan suatu metode guna menghasilkan pengenalan yang akurat. Metode yang akan digunakan dalam pengenalan citra pada aplikasi ini adalah jaringan syaraf tiruan *Learning Vector Quantization (LVQ)*.

Sistem bekerja dengan cara menangkap citra dari smartphone, kemudian mengirimkan citra tersebut ke server untuk kemudian diolah. Citra diolah melalui beberapa tahapan yaitu tahapan *image preprocessing*, tahap pendeteksian tepi dan tahap pengenalan pola citra. Dari hasil pengenalan citra tersebut, akan dikirimkan informasi singkat mengenai informasi dari objek tersebut.

Hasil uji coba yang telah dilakukan menunjukkan bahwa sistem dapat berjalan dengan baik. Sistem mampu mengirimkan citra dari smartphone ke server dan server mampu memproses citra yang akan dikenali. Hasil pengujian pengenalan citra menggunakan parameter-parameter LVQ mencakup *Epoch*, *learning rate* dan pengurangan pembelajaran diperoleh rata-rata pengenalan citra sebesar 78 %

## Kata kunci :

Aplikasi, Mobile, Citra, JST, LVQ

## 1. Pendahuluan

Kebutuhan pencarian informasi yang cepat, kapan saja dan dimana saja telah menjadi kebutuhan yang tidak bias ditawar. Kebutuhan tersebut dapat di antisipasi dengan aplikasi berbasis mobile.

Pencarian informasi yang biasanya menggunakan teks dapat diganti dengan citra. Citra dapat dijadikan

masuk untuk mencari informasi suatu objek. Penggunaan sistem pengenalan citra digunakan untuk dapat mengenali objek yang diinginkan. Jaringan syaraf tiruan – *learning vector quantization (JST-LVQ)* digunakan untuk melakukan proses pengenalan citra tersebut.

Sistem ini bertujuan untuk menguji kemampuan smartphone dalam mengirimkan citra ke server, yang kemudian citra tersebut akan diproses hingga dikenali oleh JST-LVQ. Dengan cara tersebut akan mempermudah dan mempercepat pengguna dalam melakukan pengenalan objek, sehingga pengenalan objek dapat dilakukan dimana saja dengan menggunakan smartphone.

## 2. Tinjauan Pustaka

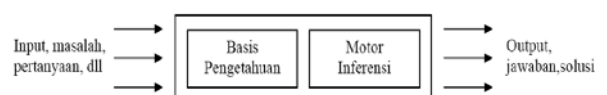
### 2.1 Kecerdasan buatan (*Artificial intelligence*)

Kecerdasan buatan (*Artificial Intelligence*) adalah bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia. [3]

Menurut John McCarthy, 1956, AI : “Untuk mengetahui dan memodelkan proses – proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.”

Cerdas adalah memiliki pengetahuan dan pengalaman, penalaran yaitu bagaimana membuat keputusan dan mengambil tindakan, serta moral yang baik. Agar mesin bisa cerdas atau bertindak seperti dan sebaik manusia, maka harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar. Penerapan kecerdasan buatan digambarkan pada Gambar 1. Dua bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan adalah :

1. Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor inferensi (*inference engine*) : kemampuan menarik kesimpulan berdasarkan pengetahuan.



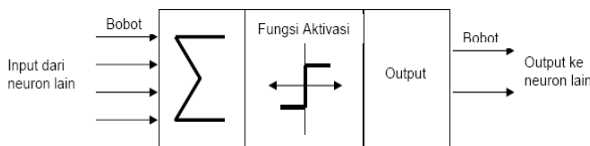
Gambar 1. Bagian Utama Kecerdasan Buatan

## 2.2 Jaringan syaraf tiruan (JST)

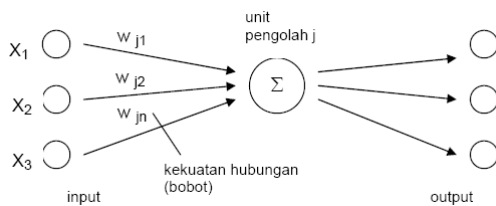
JST tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi sebagai berikut :

- Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron
- Sinyal mengalir diantara sel saraf/neuron melalui suatu sambungan penghubung
- Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini akan digunakan untuk menggandakan / mengalikan sinyal yang dikirim melaluinya.
- Setiap sel syaraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.

Model struktur neuron jaringan syaraf tiruan dijelaskan pada Gambar 2 dan Gambar 3. [5]



Gambar 2. Model Struktur JST



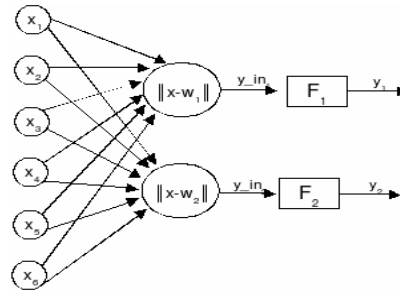
Gambar 3. Model Struktur JST

Jaringan syaraf tiruan dapat belajar dari pengalaman, melakukan generalisasi atas contoh-contoh yang diperolehnya dan mengabstraksi karakteristik esensial masukan bahkan untuk data yang tidak relevan. Algoritma untuk JST beroperasi secara langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik. JST tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, ke dalam JST dimasukkan pola-pola masukan (dan keluaran) lalu jaringan akan diajari untuk memberikan jawaban yang bisa diterima. Pada dasarnya karakteristik JST ditentukan oleh :

- Pola hubungan antar neuron (disebut arsitektur jaringan)
- Metode penentuan bobot-bobot sambungan (disebut dengan pelatihan atau proses belajar jaringan)
- Fungsi aktivasi

## 2.3 Learning vector quantization (LVQ)

LVQ merupakan jaringan syaraf dengan tipe arsitektur jaringan lapis-tunggal umpan-maju (*Single Layer Feedforward*) yang terdiri atas unit masukan dan unit keluaran. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor masukan. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor vektor masukan. Jika 2 vektor masukan mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama. Arsitektur jaringan *learning vector quantization* seperti terlihat pada Gambar 4.



Gambar 4. Arsitektur JST - Learning Vector Quantization

*Langkah 0:* Inialisasikan vektor referensi dan *learning rate*( $\alpha$ ).

*Langkah 1 :* Selama kondisi berhenti bernilai salah, kerjakan langkah 2 dan 3. (Kondisi berhenti jika perulangan mencapai maksimal perulangan dan perubahan rasio lebih kecil dari minimal eror)

*Langkah 2 :* Untuk masing-masing pelatihan vektor masukan, kerjakan:

- Temukan  $J$  sehingga  $\|x - W_j\|$  bernilai minimum.

$$C_j = \sqrt{\sum_{i=1}^n (x_i - w_{ij})^2}$$

- Perbaiki  $W_j$  dengan :
  - Jika  $T=C_j$  maka  $W_j(\text{baru}) = W_j(\text{lama}) + \alpha [x - W_j(\text{lama})]$
  - Jika  $T \neq C_j$  maka  $W_j(\text{baru}) = W_j(\text{lama}) - \alpha [x - W_j(\text{lama})]$

*Langkah 3 :* Kurangi *learning rate*.

$$\alpha = \alpha * \text{pengurang rasio}$$

*Langkah 4 :* Tes kondisi berhenti.

Keterangan rumus:

T	: Target	W	: Bobot
J	: Jumlah selisih data dan bobot	$\alpha$	: Rasio Pembelajaran
C	: Kelas selisih bobot	x	: Data

terkecil

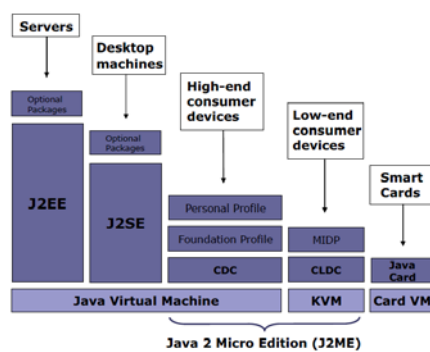
## 2.4 Perangkat Mobile

Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop. Adapun pembeda antara sistem dekstp dan perangkat mobile adalah ukurannya yang kecil, memori yang terbatas, daya proses yang terbatas, mengkonsumsi daya yang rendah dan konektifitas yang terbatas.

## 2.5 Platform JAVA

Java dibuat pada tahun 1991 oleh James Gosling. Pada awalnya diberi nama Oak, dimana untuk menghormati pohon yang ada di luar jendela Gosling. Kemudian namanya diubah ke Java karena telah ada sebuah bahasa yang diberi nama Oak. Motivasi sesungguhnya dari Java adalah kebutuhan akan sebuah bahasa yang bisa digunakan pada berbagai platform yang bisa dimasukkan ke dalam berbagai produk elektronik seperti pemanggang roti dan lemari es. Salah satu dari proyek pertama yang dikembangkan menggunakan JAVA sebuah remote kontrol yang diberi nama Star 7. [2]

Pada saat yang sama, World Wide Web dan Internet berkembang sangat cepat. Gosling menyadari bahwa Java dapat digunakan untuk pemrograman Internet. Dengan keluarnya versi 1.2, platform Java telah dipilah-pilah menjadi beberapa edisi : The Standard Edition(J2SE), Enterprise Edition(J2EE), Mobile Edition(J2ME), dan JavaCard API seperti pada Gambar 5. [4]



Gambar 5. Platform Java

## 2.6 UML (Unified Modeling Language)

UML adalah sebuah bahasa untuk menentukan, visualisasi, kontruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. [1]

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis diagram antara lain :

### a) Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara sistem dan actor. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata *user*. Sedangkan *use case* diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.

### b) Class Diagram

*Class* adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya *class* diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa *class* diagram. *Class* diagram sangat membantu dalam visualisasi struktur kelas dari suatu sistem.

### c) Component Diagram

*Component software* merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan implementasi *software* dari sebuah atau lebih *class*. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*. Sehingga component diagram merepresentasikan dunia nyata yaitu *component software* yang mengandung *component*, *interface* dan *relationship*.

### d) Deployment Diagram

Menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

### e) State Diagram

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua

class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

f) *Sequence Diagram*

*Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

g) *Collaboration Diagram*

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

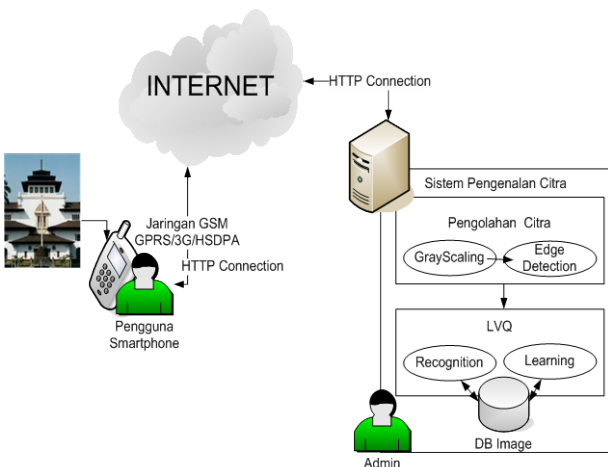
h) *Activity Diagram*

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

**3. Metode Penelitian**

**3.1 Arsitektur Sistem**

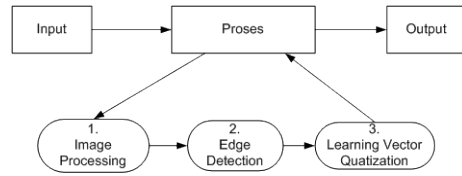
Pemanfaatan teknologi mobile dapat digunakan untuk melakukan identifikasi/pengenalan objek yang ingin diketahui. Dengan teknologi mobile suatu citra dapat dikirimkan melalui jaringan paket data (GPRS/3G/HSDPA) melalui aplikasi mobile dengan koneksi *http connection*, untuk dapat dilakukan pengenalan objek yang akan dilakukan oleh suatu server dan dikembalikan dalam bentuk informasi mengenai objek yang ingin dikenali. Bentuk arsitektur sistem yang dikembangkan dapat dilihat pada Gambar 6.



Gambar 6 Arsitektur Sistem

**3.2 Analisis Proses**

Secara umum sistem ini dibagi menjadi beberapa proses. Setiap proses ini memiliki peran masing-masing dalam menjalankan proses untuk mengenali gambar. Proses-proses tersebut tergambar pada Gambar 7



Gambar 7. Analisis Proses

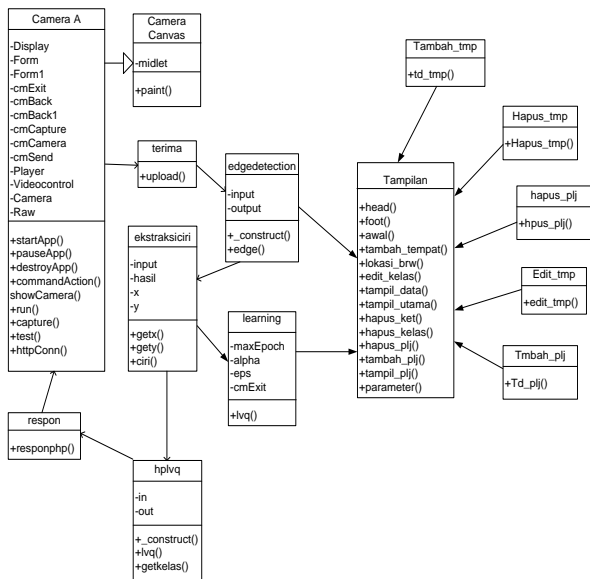
1. **Image Processing**  
Melakukan proses pengiriman gambar dari mobilephone ke server dan proses pengolahan gambar berwarna ke grayscale
2. **Edge Detection**  
Melakukan proses edge detection dengan metode Prewitt
3. **Learning Vector Quantization**  
Melakukan proses pembelajaran dengan metode Learning Vector Quantization (LVQ). Dan melakukan proses pengenalan citra dengan LVQ

**3.3 Pemodelan Sistem**

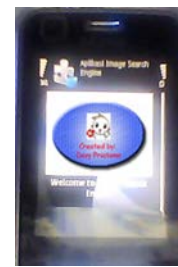
Sistem dimodelkan dengan pendekatan objek menggunakan UML. Adapun diagram yang digunakan meliputi Use case ( Gambar 8), Class (Gambar 9), dan Deployment (Gambar 10). [6]



Gambar 8. Use case diagram



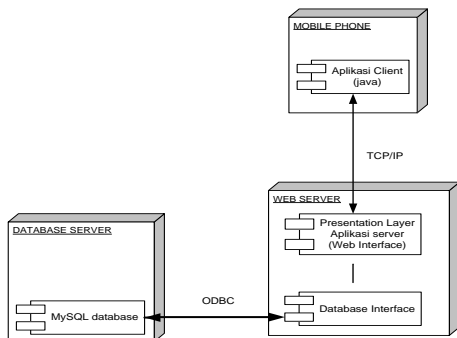
Gambar 9. Class diagram



Gambar 11. Aplikasi Mobile Pengenalan Citra



Gambar 12. Aplikasi Web Pengenalan Citra



Gambar 10. Deployment diagram

- b. Pengujian proses pelatihan citra  
 Prose pengujian pelatihan citra dilakukan dengan menggunakan citra-citra pada berukuran 160x120 pixel dan dilakukan pada aplikasi server (Gambar 13.)



Gambar 13. Data Tempat /Citra Pelatihan

Sementara citra pelatihan dapat dilihat pada Tabel 1.

Tabel 1. Citra pelatihan

Nama Tempat	Tangkuban Perahu	Gedung Sate
Citra 1		

## 4. Hasil dan Pembahasan

### 4.1 Pengujian Sistem

Pengujian sistem mencakup pengujian pengiriman gambar dari smartphone ke server, pengujian proses pelatihan (training citra) dan pengujian pengenalan citra.

- a. Pengujian pengiriman data gambar

Pengiriman data gambar dapat dilakukan dengan baik, faktor yang mempengaruhi kecepatan pengiriman gambar adalah jaringan telekomunikasi operator yang digunakan (GPRS/3G/HSDPA). Sistem berjalan dengan menggunakan 2 aplikasi yaitu aplikasi mobile (Gambar 11) dan aplikasi pada server – web pengolahan data citra (Gambar 12).



- c. Pengujian pengenalan citra  
 Pengujian pengenalan citra dilakukan dengan menggunakan parameter-parameter epoch (E), learning rate (LR) dan pengurangan pembelajaran (PB) yang beragam (Tabel 2).

Tabel 2. Pengujian parameter sistem

Parameter Pembelajaran			Presentase Pengenalan Objek	
epoch	LR	PB	Tangkuban perahu	Gedung sate
500	0.05	0.1	100%	80%
5	0.05	0.1	60%	80%
50	0.05	0.1	100%	80%
50	0.005	0.1	80%	60%
50	0.005	0.9	60%	80%

Dari hasil pengujian data maka diperoleh rata-rata akurasi pengenalan sebesar 78 %, dengan parameter pembelajaran terbaik yaitu epoch = 50, learning rate = 0.05, dan pengurangan pembelajaran = 0.1.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

1. Sistem telah mampu mengirimkan data berupa citra dari smartphone ke server dengan baik. Sementara server mampu mengolah data gambar, melakukan pengenalan objek dan memberikan jawaban.
2. Pada sistem ini, Metode JST-LVQ dapat melakukan pengenalan citra dengan rasio akurasi 78% pada kedua citra uji.
3. Metode JST-LVQ pada sistem ini mendapatkan hasil terbaik dengan menggunakan parameter pembelajaran terbaik yaitu epoch = 50, learning rate = 0.05, dan pengurangan pembelajaran = 0.1.

### 5.2 Saran

1. Pengujian dilakukan lebih banyak untuk mencari kemungkinan-kemungkinan parameter pembelajaran yang lebih baik.
2. Pengujian untuk bentuk citra-citra lain.

## Daftar Pustaka

- [1]. Ambler, Scott W. (2008). *The Elements of UML 2.0 Style*. New York: Cambridge University Press.
- [2]. Corel VENTURA™. (2003). *J2ME : The Complete Reference*. USA: McGraw - Hill,.
- [3]. Kusumadewi, Sri. (2003). *Artificial Intelegence ( Teknik Dan Aplikasinya )*, Yogyakarta : Graha Ilmu.
- [4]. Shalahuddin, M., Rosa A.S. (2008). *Pemograman J2ME : Belajar Cepat Pemograman Perangkat Telekomunikasi Mobile*, Bandung : Informatika.
- [5]. Siang, Jong Jek. (2009), *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan MATLAB*. Yogyakarta : ANDI
- [6]. Priatama, Devy. (2010), *Aplikasi Search Engine Berdasar Image Recognition dengan Jaringan Syaraf Tiruan – Learning Vector Quantization (Studi kasus pariwisata Kota Bandung)* Bandung : Skripsi UNIKOM

## Biodata Penulis

**Irawan afrianto**, memperoleh gelar Sarjana Teknik (S.T), Program Studi Teknik Informatika FTIK UNIKOM, lulus tahun 2002. Tahun 2010 memperoleh gelar Magister Teknik (M.T) dari Sekolah Teknik Elektro dan Informatika ITB.. Saat ini sebagai Staf Pengajar program studi Teknik Informatika UNIKOM.

**Devy Priatama**, Alumni Program studi Teknik Informatika FTIK UNIKOM, memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika FTIK UNIKOM, lulus tahun 2010. Bekerja sebagai Staf IT di PT. Ultrajaya Milk Industry & Trading Company, Tbk