



PETIR

JURNAL
PENGKAJIAN DAN PENERAPAN
TEKNIK INFORMATIKA

VOLUME 9 - NOMOR 1

MARET 2016

ISSN 1978-9262

PENENTUAN NASABAH PENERIMA REWARD PRODUK *GOLD* DENGAN METODE *SIMPLE ADDITIVE WEIGHTING* (SAW)
STUDI KASUS : PT. PINJAM INDONESIA

Riki Ruli A. Siregar; Faizal Fachrurrozi

IMPLEMENTASI METODE *BACKWARD CHAINING* PADA DATA *WAREHOUSE* DAOP 1 JAKARTA PT. KAI (PERSERO)

Rakhmat Arianto; Chandra Bagus Sugiarto

IMPLEMENTASI METODE DETEKSI TEPI *CANNY* PADA OBJEK SEBAGAI MODEL KEAMANAN APLIKASI PADA
SMARTPHONE ANDROID

Abdul Haris; Andi Prasetyo

ANALISA DATA DAN PERANCANGAN APLIKASI SERVICE PELANGGAN PT. JNE UNTUK PENINGKATAN KUALITAS
LAYANAN

Dewi Arianti Wulandari; Sonny Syahrindra Putra

JARINGAN AD-HOC VEHICULAR (VANET) : TINJAUAN TENTANG ARSITEKTUR, KARAKTERISTIK, APLIKASI, DAN
PROTOKOL *MEDIUM ACCESS CONTROL* (MAC)

Rosida Nur Aziza

PERANCANGAN APLIKASI PENJADWALAN MATA KULIAH MENGGUNAKAN METODE *CONSTRAINT PROGRAMMING*

Syam Gunawan

RANCANGAN TATA KELOLA PEREMAJAAN RUANG KELAS DIREKTORAT PEMBINAAN SEKOLAH DASAR

Ratna Mutu Manikam; Purwanto

PENGEMBANGAN AMORIK MENGGUNAKAN METODE GARIS SINGGUNG TERHADAP DUA LINGKARAN DAN PERSAMAAN
KURVA *BEZIER* ORDE DUA.

Darma Rusjdi

OPTIMALISASI PENGAMBILAN KEPUTUSAN PENILAIAN KINERJA DOSEN PADA PERGURUAN TINGGI ISLAM XYZ
MENGGUNAKAN *AHP* (ANALYTICAL HIERARCHY PROCESS)

Rahma Farah Ningrum

PENGAMANAN SMS PADA TELEPON SELULER BERBASIS ANDROID MENGGUNAKAN ALGORITMA *TRIPLE DES*

Raka Yusuf; M. Rival Suheri

APLIKASI LATIHAN SOAL UJIAN TEORI SURAT IZIN MENGEMUDI BERBASIS WEB

Harni Kusniyati; Raka Yusuf; Andri Setiawan

RANCANG BANGUN SIMULASI TERJADINYA LISTRIK DENGAN SUMBER DAYA SAMPAH BERBASIS MULTIMEDIA
(STUDI KASUS : TPST BANTAR GEBANG)

Yasni Djamain; Ika Fitriyani Putri

ISSN 1978-9262



771978 926272

SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)

PETIR

VOL. 9

NO. 1

HAL. 1 - 87

JAKARTA, MARET 2016

ISSN 1978-9262

IMPLEMENTASI METODE DETEKSI TEPI CANNY PADA OBJEK SEBAGAI MODEL KEAMANAN APLIKASI PADA SMARTPHONE ANDROID

Abdul Haris; Andi Prasetyo
Sekolah Tinggi Teknik – PLN, Jakarta
kemasku@yahoo.com; andi_pras@yahoo.com

ABSTRAK

Perkembangan teknologi mendorong sistem keamanan aplikasi pada smartphone android untuk mengembangkan salah satu fiturnya yaitu pengenalan objek. Pengenalan Objek adalah teknologi yang memungkinkan kita untuk mengidentifikasi atau memverifikasi objek melalui sebuah gambar digital dengan cara mencocokkan tekstur lekuk objek dengan data objek yang tersimpan di database. Misalnya mencocokkan wajah seperti lekuk hidung, mata dan dagu.

Terdapat beberapa metode untuk mendukung kerja dari pengenalan objek diantaranya yaitu deteksi tepi. Deteksi tepi dapat merepresentasikan objek-objek yang terkandung dalam citra bentuk dan ukurannya serta informasi tentang tekstur dari suatu objek. Salah satu metode pada deteksi tepi yang paling baik adalah deteksi tepi canny yang memiliki tingkat kesalahan yang minimum dibanding dengan metode deteksi tepi lain.

Deteksi tepi canny akan menghasilkan citra yang telah diolah menjadi citra baru. Citra baru tersebut akan disimpan pada basis data yang nantinya akan dicocokkan pada citra objek baru yang digunakan sebagai teknik membuka aplikasi pada smartphone android.

Kata Kunci : Sistem keamanan aplikasi, Android, Citra, Pengenalan Objek, Deteksi Tepi Canny.

ABSTRACT

The development of technology push security system applications on android smartphone to develop one of its features that is detection of the object. The detection of Objects is a technology that allows us to identify or verify an object through a digital image by matching the texture of the object with the curve of the data objects stored in the database. For example, to match the curve of the face such as the nose, eyes and chin.

There are several methods to support the work of object detection among which edge detection. Edge detection can represent the objects contained in the image of the shape and size as well as information about the texture of an object. the best method of edge detection is canny edge detection which has the minimum error rate compared with other edge detection methods.

Canny edge detection will generate the image that has been processed into a new image. The new image will be stored on a database that will be matched to the image of a new object that is used as the opening applications on android smartphone.

Keyword : Security application system, Android, Image, Object detection, Canny Edge Detection

1. Pendahuluan

Teknologi dewasa ini telah berkembang sangat pesat. Segala peralatan dan fasilitas dari berbagai bidang di dunia kini telah memanfaatkan teknologi digital dengan tujuan untuk membantu memudahkan tugas atau pekerjaan manusia. Salah satu bidang yang kini telah memanfaatkan kemajuan teknologi adalah dalam bidang telekomunikasi, terutama pada perkembangan telepon genggam

Sebagai telepon pintar atau *smartphone* yang memiliki fitur canggih didalamnya, *smartphone* membutuhkan sistem operasi yang mendukung segala kemampuan canggih tersebut, adapun sistem operasi *smartphone* yang berkembang saat ini diantaranya adalah Android, Ios, Blackberry OS, Symbian dan Windows Phone. Namun masalah yang sering muncul dari kelima sistem operasi yang berkembang saat ini adalah masalah

keamanan. Keamanan sangat dibutuhkan oleh masyarakat terutama pada masyarakat yang sangat menutamakan privasi pribadi. Sistem keamanan pada *smartphone* diutamakan untuk melindungi aplikasi penting didalamnya seperti *mobile banking*, *sms banking*, *social media* dan lain-lain sehingga *smartphone* membutuhkan pengamanan yang lebih.

Teknologi keamanan *smartphone* dengan pendeteksian objek sebenarnya telah diterapkan pada *smartphone* Android versi 4.0 Ice Cream Sandwich dengan pengenalan objek wajah/ *face recognition*. Namun terdapat banyak sekali kekurangan yang terdapat pada teknologi bawaan android tersebut seperti pengamanan *face recognition* tersebut hanya digunakan sebagai pengaman halaman utama (*Home*) saja dan bukan sebagai pengunci aplikasi tertentu yang diinginkan oleh pengguna.

Metode Deteksi tepi juga tidak dikembangkan sebagai metode pengolahan citra pada penguncian aplikasi smartphone padahal metode deteksi tepi *canny* merupakan metode yang sudah banyak digunakan dalam pendeteksian seperti pendeteksian pola batik, pendeteksian pola sarung, pendeteksian uang palsu dan lain-lain. Metode deteksi tepi *canny* juga merupakan metode terbaik dibandingkan dengan ketiga metode lainnya yaitu metode deteksi tepi *sobel*, deteksi tepi *Prewitt* dan deteksi tepi robot.

2. Landasan Teori

a. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk *smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile dan Nvidia. (Safaat H. Nazruddin, 2014)

Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaharuan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan vendor-*smartphone* sudah memproduksi *smartphone* berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Dell, Nexus, Nexian, IMO dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi android. Hal ini karena android itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

b. Deteksi Tepi *Canny*

Canny adalah algoritma deteksi tepi yang banyak digunakan dalam berbagai penelitian karena dinilai sebagai algoritma deteksi tepi yang paling optimal. Langkah awal pada algoritma *Canny* adalah mengimplementasikan tapis *Gaussian* pada citra untuk menghilangkan derau. Kemudian dilanjutkan dengan melakukan deteksi tepi pada citra dengan salah satu algoritma deteksi tepi yang ada, misalnya *Sobel* atau *Prewitt*.

Berikut adalah langkah-langkah dalam melakukan deteksi tepi *Canny* :

Langkah pertama adalah menghilangkan derau yang ada pada citra dengan menerapkan tapis *Gaussian*. Proses ini menghasilkan citra yang tampak sedikit buram. Hal ini dimaksudkan untuk mendapatkan tepian citra yang sebenarnya. Bila tidak dilakukan maka garis-garis lurus juga akan dideteksi sebagai tepian. Berikut ini adalah tapis *Gaussian*

$$\frac{1}{115}$$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Gambar 2.1 Tapis Gaussian

Selanjutnya melakukan deteksi tepi dengan salah satu operator deteksi tepi (*Operator Sobel* dan *Prewitt*) dengan melakukan pencarian secara horizontal (G_x) dan secara vertikal (G_y) melalui rumus

$$|G| = |G_x| \quad G_x = \text{pencarian secara horizontal}$$

$$G_y = \text{pencarian secara vertikal}$$

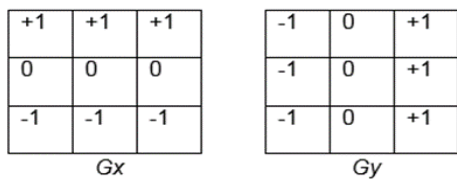
Operator Sobel yaitu sepasang kernel berupa matriks berukuran 3x3 untuk mendeteksi tepi vertikal dan horizontal.

+1	+2	+1	-1	0	+1
0	0	0	-2	0	+2
-1	-2	-1	-1	0	+1

G_x G_y

Gambar 2.2 Deteksi Tepi Vertikal dan Horizontal

Prewitt merupakan algoritma deteksi tepi yang hampir serupa dengan *Sobel*, tetapi algoritma ini menggunakan *Operator Prewitt* yang nilainya agak berbeda dengan *Operator Sobel*.



Gambar 2.3 Deteksi Tepi Vertikal dan Horizontal II

Untuk menghitung jarak *gradient*, digunakan persamaan berikut :

$$|G| = \sqrt{G_x^2 + G_y^2} \text{ Persamaan Gradient 1}$$

Kemudian untuk menghitung arah dari garis tepi yang dihasilkan, digunakan persamaan berikut :

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \text{ Persamaan Gradient 2}$$

Langkah berikutnya adalah membagi garis-garis yang ada menjadi 4 warna terpisah dengan sudut masing-masing dengan ketentuan yaitu :

1. Derajat 0 - 22,5 dan 157,5 - 180 berwarna kuning.
2. Derajat 22,5 - 67,5 berwarna Hijau.
3. Derajat 67,5 - 157,5 berwarna Merah.

Kemudian memperkecil masing-masing garis tepi agar menjadi tipis (*non maximum suppression*). Langkah terakhir adalah melakukan proses binerisasi berdasarkan nilai *low & high threshold* yang diberikan (Putra Darma,2010).

c. Flowchart

Diagram alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Diagram alir program (*program flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. (Jogiyanto, 2005).

3. Metodologi Penelitian

Pada bagian metodologi ini terdiri dari dua tahap yang pertama analisa masalah dan tahap kedua melakukan perancangan aplikasi. Berikut tahap-tahapnya :

1. Analisis Masalah

Dalam analisa ini dari gambar yang di *capture* terlebih dahulu dikonversikan ke dalam bentuk matrik seperti pada Tahap awal ini dari gambar didapatkan lalu melakukan perubahan dan pengelompokan menjadi matrik seperti berikut :



	250	240	200	200	180
	240	200	180	150	150
➔	160	180	160	150	120
	180	140	120	120	100
	160	130	100	80	60

Kemudian seleksi matrik diatas menggunakan *Gaussian* yang menghasilkan matrik :

179	171	143	143	129
171	143	129	107	107
114	129	114	107	86
129	100	86	86	71
114	93	71	57	43

Kemudian lakukan seleksi menggunakan salah satu tepian yaitu tepian *prewit* dengan matrik sebagai berikut:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

menentukan seleksi horizontal dengan menggunakan P_x, diperoleh perhitungan :

$$G_1(x,y) = |(-1 * 143) + (-1 * 114) + (-1 * 100) + (1 * 107) + (1 * 107) + (1 * 86)|$$

$$= |(-57)|$$

$$= 57$$

menentukan seleksi vertikal dengan menggunakan P_y, diperoleh perhitungan :

$$G_2(x,y) = |(-1 * 143) + (-1 * 129) + (-1 * 107) + (1 * 100) + (1 * 86) + (1 * 86)|$$

$$= |(-107)| = 107$$

$$G(x,y) = (|G_1(x,y)| + |G_2(x,y)|)$$

$$= 57 + 107 = 163$$

$$G(x,y) = \max(|G_1(x,y)| , |G_2(x,y)|) = 107$$

$$G(x,y) = (|G_1(x,y)| + |G_2(x,y)|) / 2$$

$$= (163) / 2 = 81,5 = 82$$

Langkah selanjutnya yaitu menentukan arah tepian dengan menggunakan acuan :

1. Semua arah tepi yang berkisar antara 0 dan 22,5 serta 157,5 dan 180 derajat (warna biru) diubah menjadi 0 derajat.
2. Semua arah tepi yang berkisar antara 22,5 dan 67,5 derajat (warna kuning) diubah menjadi 45 derajat.
3. Semua arah tepi yang berkisar antara 67,5 dan 112,5 derajat (warna merah) diubah menjadi 90 derajat.
4. Semua arah tepi yang berkisar antara 112,5 dan 157,5 derajat (warna hijau) diubah menjadi 135 derajat.

Berdasarkan acuan diatas maka dapat diketahui bahwa arah tepian pada citra yang telah dihitung diatas dengan nilai G(x,y) = 82 adalah 90 derajat begitu seterusnya hingga keseluruhan citra terdeteksi.

Langkah selanjutnya yaitu menetapkan fungsi *nonmaximum suspension* sehingga dihasilkan :



Setelah terbentuk citra deteksi tepi *canny* didapatkan maka dilakukan lagi mengidentifikasi matrik pada citra sehingga didapatkan :



3	4	2	5	1
2	1	6	4	2
3	5	7	1	3
4	2	5	7	1
2	5	1	3	2

Kemudian lakukan seleksi menggunakan salah satu tepian yaitu tepian *prewit* dengan matrik sebagai berikut :

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$P_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Tentukan seleksi horizontal dengan menggunakan P_x , diperoleh perhitungan :

$$G_1(x,y) = |(-1 * 3) + (-1 * 2) + (-1 * 3) + (1 * 2) + (1 * 6) + (1 * 7)|$$

$$= |(7)| = 7$$

Tentukan seleksi vertikal dengan menggunakan P_y , diperoleh perhitungan :

$$G_2(x,y) = |(-1 * 3) + (-1 * 4) + (-1 * 2) + (1 * 3) + (1 * 5) + (1 * 7)|$$

$$= |(6)| = 6$$

$$G(x,y) = (|G_1(x,y)| + |G_2(x,y)|)$$

$$= 6 + 7 = 13$$

$$G(x,y) = \max(|G_1(x,y)|, |G_2(x,y)|) = 7$$

$$G(x,y) = (|G_1(x,y)| + |G_2(x,y)|) / 2$$

$$= (6+7) / 2 = 6,5 = 7$$

Sehingga didapat nilai curmax 13 dan curmin 7.

2. Perancangan

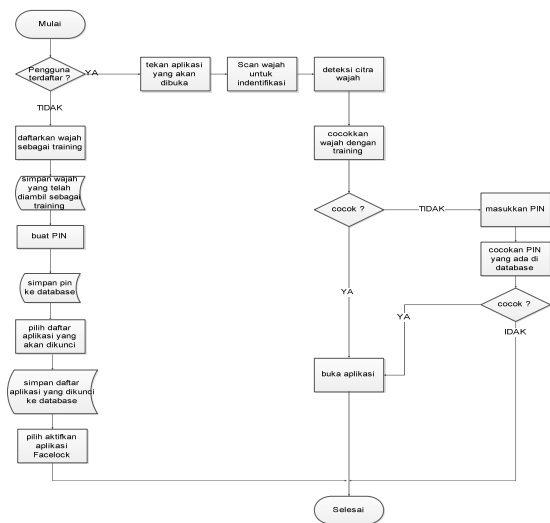
Dalam perancangan diawali dengan membuat *flowchart* untuk menggambarkan proses bisnisnya seperti berikut :

a. Diagram Flowchat

Flowchart diawali dengan kondisi apakah pengguna yang sedang menggunakan aplikasi citra wajahnya terdaftar atau tidak. Jika pengguna tidak terdaftar maka pengguna diharuskan untuk mendaftarkan wajah untuk dijadikan training

terlebih dahulu selanjutnya citra wajah tersebut disimpan ke database setelah tersimpan pengguna harus mengatur PIN kemudian setelah pin tersimpan pengguna harus memilih aplikasi yang akan dikunci oleh aplikasi "Facelock" setelah terpilih kemudian aplikasi tersebut akan tersimpan di database.

Jika pengguna yang menggunakan aplikasi merupakan pengguna terdaftar maka pengguna dapat langsung memilih aplikasi yang akan dibuka kemudian pengguna dapat menscan wajah dengan kamera. Setelah wajah di *scan* sistem akan mendeteksi citra wajah jika wajah terdeteksi sistem akan mencocokkan citra wajah dengan training yang telah tersimpan di database, jika citra wajah dengan training cocok maka aplikasi akan terbuka namun jika tidak cocok maka akan diminta PIN untuk membuka aplikasi.



b. State Transition Diagram (STD)

State Transition Diagram adalah suatu diagram yang menggambarkan bagaimana suatu proses dihubungkan satu sama lain dalam waktu yang bersamaan. Sistem yang akan digunakan merumuskan rancangan *State Transition Diagram* sebagai berikut :



3. Hasil dan Pembahasan

Pada awal akan tampil halaman buat *training* wajah akan muncul setelah pengguna membuka

aplikasi "Facelock". Sehingga tiap pengguna yang akan menggunakan aplikasi "Facelock" diharuskan untuk *training* wajah terlebih dahulu. Halaman Buat Training wajah terdiri dari *panel* kamera untuk memindai wajah pengguna. Kemudian dibawah bagian kamera terdapat tombol "tambah deteksi wajah" dimana pengguna harus memindai wajah sebanyak 7 kali.

FACELOCK



Halaman Buat PIN akan muncul setelah pengguna selesai melakukan *training* wajah setelah 7 kali melakukan *training*. Halaman PIN berfungsi untuk mengatur PIN yang digunakan jika aplikasi *Facelock* tidak dapat mendeteksi wajah. Halaman buat PIN terdiri dari nama aplikasi "Facelock" pada bagian atas layar kemudian di bawah nama aplikasi terdapat tulisan nama halaman yaitu "Atur PIN". Pada bagian tengah layar terdapat *field input* PIN dan Ulangi PIN kemudian tombol "Selesai". Setelah pengguna selesai melakukan *training* wajah dan mengatur pin maka pengguna akan masuk ke tampilan utama aplikasi. Halaman aplikasi utama memiliki dua pilihan yaitu pilihan "aktifkan aplikasi" yang berfungsi untuk mengaktifkan serta menonaktifkan aplikasi kemudian dibawah pilihan "aktifkan aplikasi" terdapat menu "Aplikasi yang dikunci" yang berfungsi untuk memilih aplikasi mana saja yang dilindungi oleh aplikasi *facelock*. Selanjutnya Menu "Aplikasi yg dikunci" berisikan daftar aplikasi yang terinstall pada perangkat android pengguna. Pada halaman ini pengguna dapat memilih aplikasi mana saja yang akan dilindungi oleh aplikasi

Facelock dengan cara menyentuh nama aplikasi yang tertera dan aplikasi akan otomatis ter-*checklist*.

Pada halaman berikutnya akan muncul ketika pengguna akan membuka aplikasi yang telah dipilih sebelumnya oleh pengguna. Halaman ini terdiri dari tulisan nama aplikasi "Facelock" yang terletak di bagian atas layar kemudian pada bagian tengah layar terdapat panel kamera yang digunakan untuk memindai wajah. Jika wajah tidak berhasil dipindai maka secara otomatis akan muncul halaman masukkan PIN. Halaman ini berisi *field input* PIN dan tombol selesai.

4. Kesimpulan dan Saran

Kesimpulan

Berdasarkan pembahasan maka pada penelitian skripsi ini didapatkan beberapa kesimpulan dan saran, yaitu:

1. Pendeteksian Objek dapat dimanfaatkan sebagai metode alternatif untuk pengunci aplikasi pada *smartphone* Android.
2. Metode Deteksi Tepi *Canny* dapat digunakan sebagai metode yang dapat diterapkan untuk keamanan pada aplikasi *smartphone* Android.

Saran

Perihal saran-saran yang dapat dijadikan bahan pertimbangan untuk pengembangan pada aplikasi yang sama yang dapat berjalan di platform IOS, blackberry, dan Windowsphone.

Daftar Pustaka

- Jogiyanto, 2005, "Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis", Penerbit Andi : Yogyakarta
- Pressman, Roger S. 2007. Rekayasa Perangkat Lunak: Pendekatan praktisi (Buku1). Martin, J. (1990). *Information Engineering: Book II – Planning and Analysis*, PrenticeHall. Penerbit Andi : Yogyakarta.
- Putra Darma, 2010, "Pengolahan Citra Digital", Penerbit Andi : Yogyakarta
- Safaat H. Nazruddin. 2012. "Pemrograman aplikasi mobile *smartphone* dan *tablet pc* berbasis *android*", Penerbit Informatika : Bandung.