[W

;/--n

**TUGAS AKHIR – KI141502**

**PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN**

**RINA WIJAYA KUSUMA WARDHANI**

**NRP 05111440000021**

**Dosen Pembimbing I**

**Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Dosen Pembimbing II**

**Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA**

**Fakultas Teknologi Informasi dan Komunikasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2018**

***[Halaman ini sengaja dikosongkan]***

**TUGAS AKHIR – KI141502**

**PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN**

**RINA WIJAYA KUSUMA WARDHANI**

**NRP 05111440000021**

**Dosen Pembimbing I**

**Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Dosen Pembimbing II**

**Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA**

**Fakultas Teknologi Informasi dan Komunikasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2018**

****

**UNDERGRADUATE THESES – KI141502**

**HUMAN ACTIVITY RECOGNITION IN VIDEO USING FEATURE COVARIANCE MATRICES**

**RINA WIJAYA KUSUMA WARDHANI**

**NRP 05111440000021**

**Supervisor I**

**Dr. Eng. Nanik Suciati, S.Kom., M.Kom.**

**Supervisor II**

**Dini Adni Navastara, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS**

**Faculty of Information and Communication Technology**

**Sepuluh Nopember Institute of Technology**

**Surabaya 2018**

***[Halaman ini sengaja dikosongkan]***

# LEMBAR PENGESAHAN

**PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

pada

Bidang Studi Komputasi Cerdas dan Visi

Program Studi S-1 Jurusan Teknik Informatika

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Oleh

**RINA WIJAYA KUSUMA WARDHANI**

**NRP : 05111440000021**

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Dr. Eng. Nanik Suciati, S.Kom., M.Kom. .....................

NIP: 197906262005012002 (Pembimbing 1)

1. Dini Adni Navastara, S.Kom., M.Sc. .....................

NIP: 198701032014041001 (Pembimbing 2)

**SURABAYA**

**JANUARI, 2018**

***[Halaman ini sengaja dikosongkan]***

**PENGENALAN AKTIVITAS MANUSIA PADA VIDEO MENGGUNAKAN FITUR MATRIKS KOVARIAN**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | **:** | **RINA WIJAYA KUSUMA WARDHANI** |
| **NRP** | **:** | **05111440000021** |
| **Jurusan** | **:** | **Informatika FTIK-ITS** |
| **Dosen Pembimbing 1** | **:** | **Dr. Eng. Nanik Suciati, S.Kom., M.Kom.** |
| **Dosen Pembimbing 2** | **:** | **Dini Adni Navastara, S.Kom., M.Sc.** |

# Abstrak

*Di zaman ini, teknologi informasi sudah dan terus berkembang secara pesat. Dengan berkembangnya teknologi informasi, berkembang pula perangkat lunak yang memudahkan masyarakat dalam berbagai bidang salah satunya bidang akademik.*

*Pada suatu penelitian terkait tentang sistem informasi akademik generik telah dihasilkan purwarupa perangkat lunak dengan fungsionalitas yang umum agar dapat digunakan pada beberapa perguruan tinggi di Indonesia. Namun, sistem generik tersebut tidak dapat menangani adanya perubahan atau perbedaan proses bisnis pada suatu perguruan tinggi. Untuk itu perlu dibangun suatu perangkat lunak yang dapat menangani hal tersebut.*

*Pada tugas akhir ini, permasalahan tersebut akan ditangani dengan membuat perangkat lunak pengintegrasi yang modular dan membuat perangkat lunak generik pada penelitian sebelumnya menjadi modul-modul yang dapat diintegrasikan ke dalam perangkat lunak pengintegrasi tersebut. Perangkat lunak pengintegrasi modul berupa web beserta kerangka kerja yang berupa abstraksi/aturan yang perlu diimplementasi modul agar dapat diintegrasi. Perangkat lunak pengintegrasi akan dapat menambah, mengganti, atau menghapus modul-modul dari perangkat lunak generik tersebut tanpa melakukan perubahan pada modul lain dan perangkat lunak pengintegrasi itu sendiri sehingga jika ada perubahan proses bisnis modul, kompleksitas untuk melakukan perubahan tersebut dapat diminimalisasi.*

***Kata kunci: Perangkat Lunak Modular, Sistem Informasi Akademik***

**HUMAN ACTIVITY RECOGNITION IN VIDEO USING FEATURE COVARIANCE MATRICES**

|  |  |  |
| --- | --- | --- |
| **Student’s Name** | **:** | **RINA WIJAYA KUSUMA WARDHANI** |
| **Student’s ID** | **:** | **05111440000021** |
| **Department** | **:** | **Informatika FTIK-ITS** |
| **First Advisor** | **:** | **Dr. Eng. Nanik Suciati, S.Kom., M.Kom.** |
| **Second Advisor** | **:** | **Dini Adni Navastara, S.Kom., M.Sc.** |

# *Abstract*

*Until today, information technology has been and will still be growing fast and because of that, many softwares that help people in many areas such as education have been developed as well.*

*In some related researches about generic academic information system, some systems that share common business processes between colleges have been developed so that those same systems can be use in many colleges in Indonesia. However, those systems cannot handle any changes and business process variation in a college that shares no common business processes with those available systems. Furthermore, those systems are still a standalone system each so that they need to be integrated into a one whole academic information system. Therefore, a software that can handle such problems is needed to be developed.*

*In this undergraduate thesis, those problems will be handled by making a modular integrator software and making those available systems into modules that can be integrated into that aforementioned modular integrator software. The aforementioned software will be a web-based application and a framework for modules to implement so that those modules can be integrated. The integrator software must be able to add, change, and delete modules without doing any code changes in another modules or the integrator itself. Therefore, if any changes or business process variations occur, the complexity to handle such problems could be minimized.*

***Keywords: Modular Software, Academic Information System***

# KATA PENGANTAR



Alhamdulillahirabbil’alamin, segala puji bagi Allah SWT, yang atas izin dan karunianya penulis dapat menyelesaikan Tugas Akhir yang berjudul ***“*RANCANG BANGUN KERANGKA KERJA SISTEM INFORMASI AKADEMIK MODULAR BERBASIS WEB DENGAN POLA ARSITEKTUR *HIERARCHICAL MODEL-VIEW-CONTROLLER”*.**

Dalam pengerjaan tugas akhir ini, penulis mendapatkan banyak sekali ilmu baru dan memperdalam ilmu-ilmu yang sebelumnya telah diajarkan selama masa perkuliahan di Teknik Informatika ITS.

Terselesaikannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ibu, Ayah, Kak Yan, Dek Farras, dan Sabrina tercinta yang senantiasa menyemangati dan menceriakan penulis selama pengerjaan tugas akhir.
3. Ibu Umi Laili Yuhana, S.Kom., M.Sc. selaku pembimbing I yang telah membantu, membimbing, dan memberikan ilmunya kepada penulis dalam menyelesaikan tugas akhir ini.
4. Bapak Rizky Januar Akbar, S.Kom., M.Eng. selaku pembimbing II yang juga telah membantu, membimbing , dan memberikan ilmunya kepada penulis dalam menyelesaikan tugas akhir ini.
5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom., M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Dosen-dosen LBE, Ibu Dr. Ir. Siti Rochimah, MT. dan Ibu Ratih Nur Esti Anggraini, S.Kom, M.Sc..
7. Teman-teman seperjuangan tugas akhir di laboratorium RPL: Rahman, Tommy, Ruslan, Galih, Bustan, Teteh, dan Ujik serta teman-teman terbaik: Ibet dan kawan-kawan yang telah menemani dan menyemangati penulis selama pengerjaan tugas akhir.
8. Adik-adik admin dan penghuni laboratorium RPL yang telah menemani penulis selama pengerjaan tugas akhir.
9. Hendy yang secara sukarela telah mebuatkan beberapa logo untuk digunakan pada perangkat lunak.
10. Pihak lain yang telah membantu penulis menyelesaikan tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2018

# DAFTAR ISI

[LEMBAR PENGESAHAN v](#_Toc515489547)

[Abstrak vii](#_Toc515489548)

[*Abstract* ix](#_Toc515489549)

[KATA PENGANTAR xi](#_Toc515489550)

[DAFTAR ISI xiii](#_Toc515489551)

[DAFTAR GAMBAR xvi](#_Toc515489552)

[DAFTAR TABEL xvii](#_Toc515489553)

[DAFTAR KODE SUMBER xviii](#_Toc515489554)

[BAB I PENDAHULUAN 1](#_Toc515489555)

[1.1 Latar Belakang 1](#_Toc515489556)

[1.2 Rumusan Masalah 2](#_Toc515489557)

[1.3 Batasan Masalah 3](#_Toc515489558)

[1.4 Tujuan 3](#_Toc515489559)

[1.5 Manfaat 4](#_Toc515489560)

[1.6 Metodologi 4](#_Toc515489561)

[1.6.1 Penyusunan Proposal Tugas Akhir 4](#_Toc515489562)

[1.6.2 Studi Literatur 4](#_Toc515489563)

[1.6.3 Analisis dan Desain Perangkat Lunak 5](#_Toc515489564)

[1.6.4 Implementasi Perangkat Lunak 5](#_Toc515489565)

[1.6.5 Pengujian dan Evaluasi 5](#_Toc515489566)

[1.7 Sistematika Penulisan Laporan Tugas Akhir 6](#_Toc515489567)

[BAB II DASAR TEORI 9](#_Toc515489568)

[2.1 Pengenalan Aktivitas Manusia 9](#_Toc515489570)

[2.2 *Optical Flow* 10](#_Toc515489571)

[2.3 Fitur Matriks Kovarian 12](#_Toc515489572)

[2.4 Normalisasi 14](#_Toc515489573)

[2.5 Matriks Log-Kovarian 14](#_Toc515489574)

[2.6 Nearest-Neighbor (NN) Classification 14](#_Toc515489575)

[2.7 *Confusion Matrix* 15](#_Toc515489576)

[BAB III PERANCANGAN PERANGKAT LUNAK 16](#_Toc515489577)

[3.1 Lingkungan Desain dan Implementasi 16](#_Toc515489580)

[3.2 Perancangan Sistem 16](#_Toc515489581)

[3.3 Perancangan Data 19](#_Toc515489582)

[3.4 Perancangan Proses 20](#_Toc515489583)

[3.4.1 *Preprocessing* 20](#_Toc515489584)

[3.4.2 Koleksi Fitur Lokal 21](#_Toc515489585)

[3.4.3 Representasi Aktivitas 22](#_Toc515489586)

[3.4.4 Klasifikasi 23](#_Toc515489587)

[BAB IV IMPLEMENTASI 25](#_Toc515489588)

[4.1 Lingkungan Pengembangan 25](#_Toc515489590)

[4.2 Implementasi Preprocessing 25](#_Toc515489591)

[4.2.1 *Web* *layer* 25](#_Toc515489592)

[4.3 Uji Coba 25](#_Toc515489593)

[BAB VI KESIMPULAN DAN SARAN 26](#_Toc515489594)

[5.1 Kesimpulan 26](#_Toc515489596)

[5.2 Saran 26](#_Toc515489597)

[DAFTAR PUSTAKA 27](#_Toc515489598)

[BIODATA PENULIS 31](#_Toc515489599)

***[Halaman ini sengaja dikosongkan]***

# DAFTAR GAMBAR

***[Halaman ini sengaja dikosongkan]***

# DAFTAR TABEL

***[Halaman ini sengaja dikosongkan]***

# DAFTAR KODE SUMBER

***[Halaman ini sengaja dikosongkan]***

# BAB I PENDAHULUAN

## Latar Belakang

Saat ini, pengenalan aktivitas manusia pada data video merupakan persoalan yang cukup menantang di bidang visi komputer. Hal ini disebabkan oleh kompleksitas pada data video berupa halangan, kekacauan, interaksi objek ganda, perubahan cahaya, dan lain-lain. Selain itu, adanya persoalan pada proses pengambilan video seperti penyimpangan, pergerakan, dan sudut pandang kamera, serta kompleksitas pada aktivitas manusia itu sendiri sebagai objek non-rigid yang memiliki beragam kelas aktivitas [1]. Pengenalan aktivitas manusia pada data video telah diterapkan di beberapa bidang serta digunakan sebagai sistem keamanan atau sarana hiburan. Untuk itulah penulis ingin menerapkan pengenalan aktivitas manusia pada data video yang berasal dari *Closed-Circuit Television* (CCTV) yang berada di Departemen Informatika ITS Surabaya.

Ada dua komponen dasar pada setiap algoritma pengenalan aktivitas yang memengaruhi hasil akurasi dan efisiensi yaitu representasi model aktivitas dan metode klasifikasi yang digunakan. Model yang digunakan untuk merepresentasikan aktivitas dapat dibagi menjadi lima kategori yaitu *shape models, motion models, geometric human body models, interest-point models, dan dynamic models*. Metode klasifikasi standar yang paling sering digunakan antara lain *nearest-neighbor* (NN) classifier, *support vector machine* (SVM), dan *boosting* [1].

Pengenalan aktivitas dengan *shape-based models* menggunakan estimasi akurasi dari siluet objek yang bergerak di setiap *frame* video sehingga membentuk *silhouette tunnel*. Model ini bersifat invarian terhadap pencahayaan, warna, dan tekstur dari objek yang bergerak, namun tidak semua data video memiliki model ini [1-10]. Pengenalan aktivitas dengan *geometric human body models* menggunakan bagian tubuh manusia sebagai modelnya sehingga akan sulit didapat dari data video dengan objek yang kecil dan *background* yang luas [11-14]. Pengenalan aktivitas dengan *interestpoint models* dilakukan dengan mengekstrak *intereset-point* dari pertimbangan informasi struktural dan mendeteksi *cuboids* di daerah yang memiliki kemungkinan besar mengalami pergerakan [15-20]. *Dynamic models* merupakan model yang paling baru digunakan dalam pengenalan aktivitas [21], [22]. Model ini mendeskripsikan postur statis dari suatu aksi sebagai sebuah *state* dan mendeskripsikan *dynamics* (variasi temporal) dari suatu aksi menggunakan transisi model *state-space*. Model yang terakhir yaitu *motion models*. Model ini mengekstrak berbagai karakteristik pergerakan dan deformasi objek. Model ini menggunakan fitur kinematic yang berasal dari *optical flow* untuk merepresentasikan aktivitas [23-30]. *Motion-model* dikatakan sebagai atribut yang paling diskriminatif dalam merepresentasikan aksi [1].

Komponen dasar yang akan digunakan dalam pembuatan tugas akhir ini adalah penggunaan fitur matriks kovarian sebagai representasi model aktivitas. Fitur matriks kovarian merupakan representasi ringkas dari koleksi fitur lokal sehingga memiliki dimensi yang lebih sedikit. Koleksi fitur lokal yang digunakan berasal dari *optical flow* data video yang termasuk ke dalam *motion-model*. Komponen dasar lainnya adalah metode klasifikasi aktivitas yang digunakan yaitu *nearest-neighbor* (NN) *classifier* yang diketahui tahan terhadap keanekaragaman bentuk aktivitas, perubahan sudut pandang kamera, dan objek dengan resolusi rendah. Secara konseptual, *framework* ini cukup sederhana, membutuhkan ruang penyimpanan yang kecil, serta memiliki perhitungan yang memungkinkan untuk implementasi secara real time [1].

## Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

Bagaimana cara mendapatkan koleksi fitur lokal berbasis *optical flow* dari data video?

Bagaimana cara mendapatkan representasi model aktivitas manusia menggunakan fitur matriks kovarian?

Bagaimana cara klasifikasi aktivitas manusia menggunakan *nearest-neighbor* (NN) *classifier*?

Bagaimana mengetahui kinerja dari aplikasi pengenalan aktivitas manusia pada data video?

## Batasan Masalah

Permasalan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

Data yang digunakan pada tugas akhir ini berupa data video yang diperoleh dari CCTV di Departemen Informatika ITS Surabaya.

Karena kompleksitas dan keberagaman aktivitas yang ada, pengenalan aktivitas hanya berupa aktivitas yang berasal dari objek tunggal.

Meskipun dapat digunakan untuk mengenali aktivitas berbagai macam jenis objek, tugas akhir ini hanya berfokus pada aktivitas manusia.

Aktivitas yang dapat dikenali berjumlah tiga yaitu aktivitas berlari, berjalan, dan melambaikan tangan,

## Tujuan

Tujuan dari tugas akhir ini adalah menciptakan aplikasi pengenalan aktivitas manusia pada video yang terekam oleh CCTV.

## Manfaat

Manfaat tugas akhir ini adalah sebagai langkah awal dalam membangun sistem pengawasan dan keamanan di Departemen Informatika ITS Surabaya melalui data rekaman video CCTV.

## Metodologi

### Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini akan mendeskripsikan dan membahas mengenai rencana pembuatan aplikasi pengenalan aktivitas manusia pada data video CCTV. Secara detil, proposal tugas akhir ini berisi tentang beberapa bagian yaitu latar belakang diajukannya tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir dan ringkasan isi yang membahas metode yang akan digunakan dalam tugas akhir. Sub bab metodologi merupakan penjelasan mengenai tahapan penyusunan tugas akhir. Terdapat pula sub bab jadwal pengerjaan yang menjelaskan jadwal pengerjaan tugas akhir dan di akhir bagian terdapat daftar pustaka untuk mencantumkan referensi yang digunakan dalam tugas akhir.

### Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang metode-metode yang akan digunakan dalam pengenalan aktivitas manusia pada data video. Informasi dan studi literatur tersebut didapat dari buku, internet, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

### Analisis dan Desain Perangkat Lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dilakukan desain sistem dan desain proses-proses yang ada.

### Implementasi Perangkat Lunak

Pengembangan aplikasi dalam tugas akhir nantinya akan menggunakan matlab sebagai compiler dan library yang telah tersedia di matlab.

### Pengujian dan Evaluasi

Pada tahapan ini dilakukan uji coba terhadap aplikasi yang telah dibuat. Pengujian ini betujuan untuk mengetahui unjuk kerja dari aplikasi pengenalan aktivitas pada data video. Pada uji coba ini, langkah pertama adalah melatih sistem dengan data training untuk membentuk model aktivitas. Langkah selanjutnya akan diberikan data uji untuk menguji model aktivitas yang telah terbentuk serta menghasilkan prediksi jenis aktivitas dari data uji. Parameter yang akan digunakan sebagai bahan uji coba adalah ukuran dari overlapping frame yang digunakan pada proses preprocessing video.

Setelah uji coba, dilakukan juga evaluasi terhadap kinerja dari aplikasi ini dengan cara melakukan pengukuran accuracy dan f-measure (precision dan recall) dari hasil uji coba yang berupa prediksi aktivitas terhadap aktivitas sebenarnya. Perhitungan accurac dan f-measure dapat dilihat pada Tabel 1 dan persamaan (1.1), (1.2), dan (1.3). Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan perencanaan. Tahap ini dimaksudkan untuk mengetahui kinerja aplikasi, mengevaluasi jalannya sistem, mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **Nilai Sebenarnya** | |
| ***TRUE*** | ***FALSE*** |
| **Nilai Prediksi** | ***TRUE*** | (*True Positive*) *Correct Result* | (*False Positive*) *Unexpected Result* |
| ***FALSE*** | (*False Negative*) *Missing Result* | (*True Negative*) *Correct Absence of Result* |

(1.1)

(1.2)

(1.3)

## Sistematika Penulisan Laporan Tugas Akhir

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini:

**Bab I Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

**Bab III Perancangan Perangkat Lunak**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

**Bab IV Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

**Bab V Uji Coba dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI Kesimpulan dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

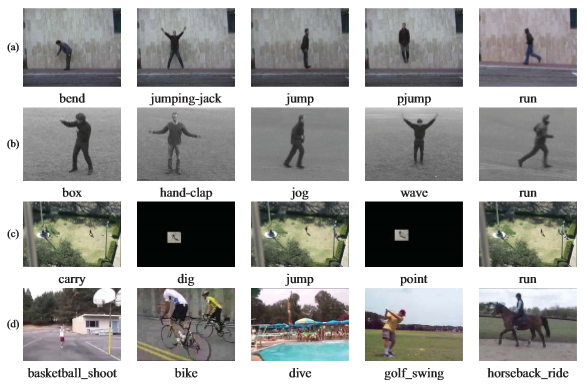
***[Halaman ini sengaja dikosongkan]***

# BAB II DASAR TEORI



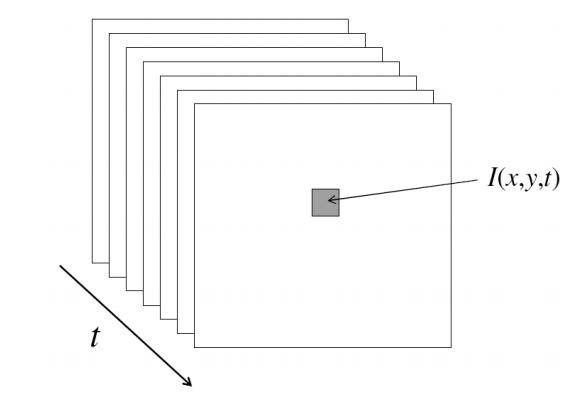
## Pengenalan Aktivitas Manusia

Pengenalan aktivitas manusia dapat dikelompokkan kedalam beberapa jenis. Pertama, pengenalan pada aktivitas sederhana yang dilakukan manusia seperti berjalan, naik tangga, turun tangga, jogging, dan lain-lain. Kedua, pada aktivitas kompleks yang biasanya mengkombinasikan aktivitas menjadi aktivitas dengan waktu yang lama, seperti; menunggu bus, mengemudi, dan lain-lain. Sebuah aktivitas juga dapat dilakukan hanya dengan beberapa anggota tubuh seperti; mengetik, melambaikan tangan, dan lain-lain. Aktivitas manusia juga dapat dikenali dengan menggunakan sensor seperti Accelerometer, Gyroscope, Camera, dan GPS [2]. Contoh aktivitas manusia pada data video ditunjukkan pada Gambar 1.



## *Optical Flow*

*Optical flow* merupakan fitur lokal yang dapat menangkap gerakan dinamis yang merupakan karakteristik aktivitas manusia dari data video dan terdiri dari 12 dimensi fitur. Pada beberapa tahun belakangan ini, sudah banyak penelitian yang membahas mengenai perhitungan *optical flow*. Pada kasus ini, yang akan digunakan adalah varian dari metode *Horn and Schunck*, yang mengoptimalkan fungsional berdasarkan residu dari batasan intensitas dan persyaratan regularisasi kehalusan [3].

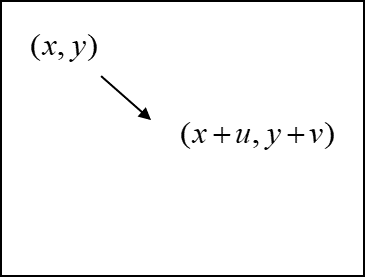


menunjukkan intensitas pencahayaan pada rangkaian video di posisi piksel seperti ilustrasi pada Gambar 2. mewakili vector optical flow yang sesuai, dimana **.** Berdasarkan dan dapat didefinisikan vector fitur lokal pada persamaan (2.1).

(2.1)

dimana (himpunan semua koordinat piksel pada segmen video), adalah turunan parsial pertama dari yang berkenaan dengan waktu , sehingga dapat didefinisakan seperti pada persamaan (2.2).

(2.2)



Gambar . Ilustrasi Perpindahan Intensitas pada *Frame*

dan adalah komponen *optical flow* yang merupakan perpindahan posisi piksel dengan intensitas I berdasarkan seperti ilustrasi pada Gambar 3. Diasumsikan intensitas piksel antara *frame* satu dengan *frame* lainnya bersifat konstan, sehingga dapat didefinisikan seperti pada persamaan (2.3). sementara dan adalah turunan parsial pertamanya yang berkenaan dengan .

(2.3)

(*divergence*) adalah perbedaan spasial dari *flow field* yang ada pada setiap posisi piksel. *Divergence* menangkap jumlah ekspansi lokal yang dapat mengindikasikan perbedaan aktivitas. *Divergence* didefinisikan pada persamaan (2.4).

(2.4)

(*voricity*) digunakan untuk mengukur putaran lokal di sekitar sumbu yang tegak lurus terhadap bidang *flow field*. Pada konteks *optical flow*, *voricity* berpotensi menangkap gerakan melingkar lokal dari objek yang bergerak. *Voricity* didefinisikan pada persamaan (2.5).

(2.5)

Untuk menggambarkan dan ada dua matriks yang berperan, yaitu tensor gradien dari *optical flow* dan laju regangan tensor yang didefinisikan pada berurutan pada persamaan (2.6) dan (2.7).

(2.6)

(2.7)

dan adalah invarian tensor yang tetap konstan tidak peduli sistem koordinat apa yang mereka rujuk. dan juga merupakan sifat skalar yang menggabungkan komponen tensor gradien sehingga menghasilkan struktur lokal. dan didefinisikan berurutan pada persamaan (2.8) dan (2.9).

(2.8)

(2.9)

dimana menunjukkan *trace operation*.

## Fitur Matriks Kovarian

Sampel video biasanya berdimensi sangat tinggi. Tentu sangat tidak praktis mempelajari struktur global sampel video *learning* dan membangun klasifikasi secara langsung di ruang berdimensi tinggi. Oleh karena itu, dilakukanlah pendekatan pemodelan “*bag of dense local feature vector*” dimana serangkaian fitur lokal diekstrak dari data video untuk merepresentasikan aktivitas. Namun, dimensi sekumpulan vector fitur lokal tersebut bahkan lebih besar daripada sampel video yang diekstraksi karena jumlah piksel yang dikali dengan ukuran vector fitur tersebut. Sehingga diperlukannya pengurangan dimensi.

Fitur matriks kovarian dapat menyediakan representasi yang sangat baik untuk pengenalan aktivitas. Selain sederhana dan efektif, matriks kovarian dari fitur lokal memiliki ruang penyimpanan dan kebutuhan proses yang kecil. [4] [5]

menunjukkan “*bag of feature vector*” dan adalah ukuran dari . Matriks kovarian dari dapat didefinisikan seperti pada persamaan (2.10).

(2.10)

dimana adalah mean dari vector fitur yang didefinisikan pada persamaan (2.11).

(2.11)

Matriks kovarian menyediakan cara alami untuk menggabungkan beberapa vector fitur. Dimensi dari matriks kovarian hanya berhubungan dengan dimensi dari vector fitur. Jika berdimensi , maka merupakan matriks berukuran . Karena jauh lebih kecil dari , matriks akan memiliki dimensi yang jauh lebih kecil dibandingkan dengan dimensi dari “*bag of feature vector*” yang membutuhkan dimensi.

## Normalisasi

Normalisasi merupakan teknik untuk menstandarkan atau menyamakan rentang data sehingga tidak ada satu atribut yang terlalu dominan atas atribut yang lain. Salah satu teknik normalisasi adalah normalisasi skala.

Normalisasi skala merupakan teknik penstandaran data pada suatu rentang tertentu, umumnya 0-1. Diketahui nilai maksimal dari suatu piksel citra sebagai 𝑁𝑚𝑎𝑥 dan nilai minimal dari suatu piksel citra adalah 𝑁𝑚𝑖𝑛. Citra yang dinormalisasi disimbolkan dengan N. Nilai hasil normalisasi didapatkan dari rumus perhitungan normalisasi skala ditunjukkan pada persamaan (2.12)

(2.12)

## Matriks Log-Kovarian

Himpunan dari semua matriks kovarian dengan ukuran tertentu tidak membentuk sebuah ruang vector karena tidak diikuti oleh perkalian dengan scalar negatif, sehingga membentuk “*closed convex cone*” [6]. Kebanyakan dari algoritma *machine learning* bekerja dengan fitur yang berada pada ruang Euclidean, bukan *convex cone.* Oleh karena itu, diperlukannya pemetaan *convex cone* pada matriks kovarian ke dalam ruang vector dengan menggunakan logaritma dari matriks kovarian [7]. Perhitungan matriks kovarian dijelaskan pada persamaan (2.13).

(2.13)

dimana adalah *orthonormal eigenvectors* dan adalah matriks diagonal dari *eigenvalues*. Selanjutnya didefinisikan seperti pada persamaan (2.14).

(2.14)

Dimana adalah matriks diagonal dari matriks dengan mengganti entri diagonal matriks dengan nilai logaritmanya.

## Nearest-Neighbor (NN) Classification

Nearest-neighbor classification merupakan algoritma yang paling banyak digunakan untuk klasifikasi. Algoritma ini cukup sederhana dan mudah; diberikan sebuah sampel data testing. Temukan sampel yang paling mirip dengan sampel data testing tersebut pada himpunan data training, dimana kemiripan diukur dari suatu perhitungan jarak antara keduanya. Selanjutnya, beri label kelas pada sampel data testing tersebut.

Keberhasilan dari klasifikasi menggunakan NN classifier bergantung pada metrik jarak yang digunakan. Pada kasus ini, metric yang akan digunakan adalah pengukuran jarak Euclidean antara logaritma dari representasi aktivitas matriks kovarian.

(2.15)

dimana adalah logaritma matriks kovarian dan menunjukkan norma *Frobenius* pada matriks. Jarak yang didefinisikan pada persamaan (2.15) dapat ditunjukkan sebagai metrik *Riemannian* untuk jenis matriks kovarians. Metrik Ini disebut sebagai metrik *log-Euclidean*.

## *Confusion Matrix*

Confusion matrix merupakan matriks yang mengandung informasi tentang kelas sebenarnya dan prediksi yang dihasilkan oleh sistem klasifikasi. Confusion matrix banyak digunakan untuk menguji performa dari suatu metode klasifikasi. Struktur confusion matrix untuk tiga kelas ditunjukkan pada Tabel 2.1.

Table 2. Confusion Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Nilai Prediksi** | | |
| ***1*** | ***2*** | ***3*** |
| **Nilai Sebenarnya** | ***1*** |  |  |  |
| ***2*** |  |  |  |
| ***3*** |  |  |  |

Keterangan:

= jumlah data uji yang kelas prediksinya sama

dengan kelas yang sebenarnya.

= jumlah kelas prediksi yang tidak sesuai dengan

kelas sebenarnya.

Beberapa nilai evaluasi yang bisa dihitung berdasarkan confusion matrix untuk mengetahui performa *classifier* yaitu *accuracy, recall,* dan *precision*. *Accurac*y adalah perbandingan jumlah total data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap seluruh data. Rumus perhitungan dari akurasi ditunjukkan pada persamaan (2.16).

(2.16)

*Recall* adalah perbandingan dari jumlah data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap seluruh data berkelas benar pada ground truth. Rumus perhitungan Sensitivity ditunjukkan pada persamaan (2.17), (2.18), (2.19).

(2.17)

(2.18)

(2.19)

*Precision* adalah adalah perbandingan dari jumlah data yang prediksi kelas hasil klasifikasinya sesuai dengan *ground truth* terhadap keseluruhan data yang terklarifikasi benar. Rumus perhitungan precision ditunjukkan pada persamaan (2.20), (2.21), (2.22).

(2.20)

(2.21)

(2.22)

***[Halaman ini sengaja dikosongkan]***

# BAB III PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dibahas perancangan aplikasi pengenalan aktivitas manusia pada video. Perancagan yang dibuat meliputi perancangan sistem, perancangan data, dan perancangan proses. Hasil dari proses ini berupa diagram yang akan digunakan sebagai acuan untuk proses implementasi perangkat lunak.



## Lingkungan Desain dan Implementasi

Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam desain dan implementasi pembuatan aplikasi disebutkan dalam Tabel 3.1.

Table 3.1 Lingkungan Perancangan Perangkat Lunak

|  |  |  |
| --- | --- | --- |
| **No.** | **Jenis** | **Spesifikasi** |
| 1 | Prosesor | Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz 3.40 GHz |
| 2 | Memori | 4.00 GB |
| 3 | Sistem Operasi | Windows 10 Enterprise 64-bit |
| 4 | Perangkat Lunak | Matlab R2018a  Ms. Excel 2016  Ms.Word 2016 |

## Perancangan Sistem

Perancangan sistem dilakukan untuk menggambarkan proses secara keseluruhan dari aplikasi pengenalan aktivitas manusia pada video secara keseluruhan. Rancangan sistem aplikasi ini ditunjukan dalam bentuk diagram alir pada Gambar x yang terdiri dari satu data masukan, empat tahapan proses, dan satu data keluaran.

Gambar 3. Diagram Alir Perancangan Sistem

Preprocessing

Koleksi FiturLokal

Representasi Aktivitas

Klasifikasi

Video

Jenis Aktivitas

Start

Finish

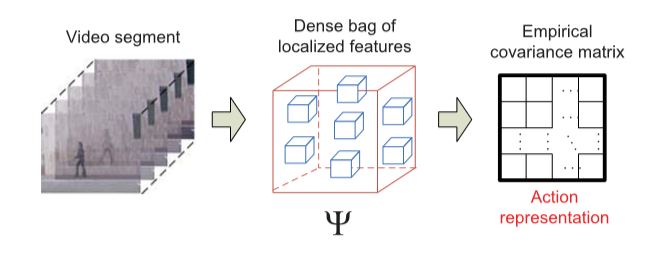
Data masukan yang digunakan adalah video *CCTV* yang berada di Departemen Informatika ITS lantai 3, sebagai salah satu tempat yang paling sering terjadi aktivitas manusia seperti berjalan atau berlari. Contoh potongan video *CCTV* tersebut ditunjukkan pada Gambar 3.1.



Gambar 3. View CCTV Departemen Informatika ITS

Proses *preprocessing* adalah proses untuk menyiapkan citra video sebelum masuk ke proses pengumpulan fitur. Persiapan tersebut berupa pengumpulan video berdasarkan aktivitas, *resize* video, mengubah video menjadi citra *greyscale,* dan memotong video menjadi segmen-segmen kecil yang mewakili suatu aktivitas. Hasil segmen tersebut digunakan sebagai data masukan proses koleksi fitur.

Proses koleksi fitur lokal adalah pengumpulan fitur-fitur lokal pada tiap segmen video berdasarkan fitur *optical flow*-nya. Fitur-fitur lokal tersebut terdiri dari 12 dimensi fitur yang diambil dari setiap piksel citra video. Sehinga hasil dari proses ini berupa *bag of localized features* berukuran N x 12 dimana N adalah jumlah seluruh piksel dalam satu segmen video. *bag of localized features* inilah yang akan menjadi data masukan proses representasi aktivitas.

Seperti ilustrasi pada Gambar 3.2, setelah fitur-fitur lokal (*bag of localized features*) dari segmen video terkumpul, selanjutnya adalah merepresentasikannya menjadi sebuah aksi atau aktivitas. Proses representasi aktivitas adalah proses pengekstrakan fitur-fitur lokal menjadi representasi aktivitas menggunakan fitur matriks kovarian. Hal ini bertujuan untuk mendapatkan dimensi yang jauh lebih kecil.

Gambar 3. Ilustrasi Perubahan Dimensi Video

Setelah diperoleh fitur matriks kovariannya, dilanjutkan dengan menghitung logaritma dari matriks kovarian tersebut. Hal ini bertujuan untuk memetakan bentuk matriks yang sebelumnya “*convex cone*” menjadi bentuk ruang vector sehingga dapat digunakan oleh *machine learning* untuk proses klasifikasi selanjutnya.

Proses klasifikasi dilakukan menggunakan Nearest-Neighbor classifier dengan jarak Euclidean sebagai distance metricnya.

## Perancangan Data

Perancangan data dilakukan untuk memastikan pengoperasian aplikasi berjalan dengan benar. Data masukan (input) adalah data yang diperlukan dalam pengoperasian aplikasi dan data keluaran (output) adalah data yang dihasilkan oleh aplikasi serta akan digunakan oleh pengguna.

Data masukan pada aplikasi pengenalan aktivitas manusia adalah data video berukuran 636x360 piksel dengan citra warna RGBdan memiliki *framerate* sebesar 25 *fps*. Panjang frame yang akan diproses tiap iterasi adalah sebanyak 8 dan 20 frame karena merupakan panjang frame yang paling ideal dalam merepresentasikan suatu gerakan aktivitas.

Satu data video menampilkan satu objek yang melakukan satu jenis aktivitas. Aktivitas yang dilakukan yaitu aktivitas berlari, berjalan, dan melambaikan kedua tangan. Masing-masing aktivitas tersebut terdiri dari beberapa ragam pola arah dan titik gerakan pada latar video. Seperti yang ditunjukkan pada Gambar 3.3, ada 6 arah aktivitas lari, 6 arah aktivitas jalan, dan 4 titik aktivitas melambaikan kedua tangan. Contoh potongan video *CCTV* yang akan menjadi data masukan ditunjukkan pada Gambar x.

Data keluaran aplikasi pengenalan aktivitas manusia merupakan prediksi aktivitas yang dilakukan objek pada data masukan baik itu lari, jalan, atau melambaikan kedua tangan.



Gambar 3. Arah dan Titik Gerak Aktivitas pada CCTV

## Perancangan Proses

Perancangan proses dilakukan untuk memberikan gambaran mengenai setiap proses yang terdapat pada aplikasi pengenalan aktivitas manusia. Bagian dari setiap proses utama aplikasi dapat dilihat pada Gambar 3.1.

### *Preprocessing*

Proses pertama yang dilakukan adalah *preprocessing* yaitu mengubah ukuran video yang sebelumnya 1080x720 piksel diperkecil menjadi 636x360 piksel dan mengubah citra video dari RGB menjadi *greyscale*. Hal tersebut dilakukan untuk mengoptimalkan kerja aplikasi. Diagram alir proses ini ditunjukkan pada Gambar 3.5.

Gambar 3. Diagram Alir Proses *Preprocessing*

Ubah ukuran video

Ubah citra video menjadi *greyscale*

Video

Citra video hasil p*reprocessing*

Start

Finish

### Koleksi Fitur Lokal

Proses koleksi fitur lokal adalah pengumpulan fitur-fitur lokal pada citra hasil *preprocessing* berdasarkan fitur *optical flow*-nya. Fitur-fitur lokal tersebut terdiri dari 12 dimensi fitur yang diambil dari setiap piksel citra video.

Gambar 3. Diagram Alir Proses Koleksi Fitur Lokal

Matriks Fn

Hitung fitur Div, Vor, Gten, Sten

Finish

Normalisasi

Inisialisasi fitur It, u, v

Gabung ke-12 fitur menjadi matriks Fn

Hitung fitur ut dan vt

Hitung fitur It, u, v

Ambil fitur x, y, t

Citra hasil *preprocessing*

Start

Fitur u dan v dari citra sebelumnya

t % L = 1

t % L = 0

yes

yes

no

no

Fitur-fitur tersebut antara lain . Fitur-fitur tersebut akan dikumpulkan menjadi satu matriks Fn setiap frame sehingga akan memiliki ukuran dimensi sebesar Diagram alir proses ini ditunjukkan pada Gambar 3.6.

Proses pertama yaitu mengambil fitur yang menunjukkan posisi-posisi piksel pada citra masukkan. Selanjutnya adalah pengambilan fitur dengan terlebih dahulu mengecek apakah citra masukkan merupakan citra pertama atau bukan. Jika citra masukkan termasuk citra pertama, maka akan dilakukan inisialisasi fitur . Jika bukan, maka akan dilakukan perhitungan fitur tersebut menggunakan data fitur dari citra sebelumnya.

Proses selanjutnya adalah perhitungan fitur yang merupakan turunan parsial masing-masing fitur terhadap posisi frame . Dilanjutkan dengan perhitungan fitur menggunakan data fitur .

Selanjutnya, ke-12 fitur masing-masing piksel pada citra digabungkan menjadi sebuah matriks Fn. Setelah itu dilakukan pengecekan apakah citra masukan tersebut merupakan citra terakhir atau bukan. Jika citra masukan bukan termasuk citra terakhir, maka proses koleksi fitur lokal selesai dilanjutkan dengan koleksi fitur lokal pada citra selanjutnya sampai dengan citra terakhir pada frame ke-L.

Setelah terkumpul fitur-fitur lokal pada citra-citra sepanjang L frame dalam bentuk matriks Fn, proses terakhir adalah normalisasi matriks tersebut menggunakan normalisasi skala.

### Representasi Aktivitas

Proses Representasi aktivitas adalah proses pengekstrakan matriks Fn menjadi suatu data yang dapat digunakan untuk merepresentasikan sebuah aktivitas menggunakan matriks kovariannya. Hal tersebut dilakukan untuk mengurangi ukuran dimensi matriks Fn menjadi jauh lebih kecil yaitu dari ukuran menjadi . Diagram alir proses ini ditunjukkan pada Gambar 3.7.

Proses pertama adalah dengan menghitung mean matriks Fn terhadap ke-12 fitur optical flow, dilanjutkan dengan menghitung mariks kovariannya.

Proses selanjutnya adalah menghitung logaritma dari matriks kovarian. Hal ini dilakukan untuk membentuk matriks menjadi suatu ruang vector sehingga dapat digunakan oleh proses selanjutnya yakni klasifikasi.

Gambar 3. Diagram Alir Proses Representasi Aktiviats

Hitung mean Fn

Hitung matriks covariance

Matriks Fn

Matriks log covariance

Start

Finish

Hitung matriks log covariance

### Klasifikasi Akivitas

Proses terakhir adalah proses klasifikasi untuk mengetahui jenis aktivitas yang dilakukan menggunakan Nearest-Neighbor classifier. Diagram alir proses ini ditunjukkan pada Gambar 3.8.

Gambar 3. Diagram Alir Proses Klasifikasi

Buat model training NN *Classifier*

Hitung jarak euclidean

Matriks log covariance

Hasil aktivitas

Start

Finish

Cari jarak terpendek

Membentuk model training Nearest-Neighbor classifier dilakukan untuk melatih aplikasi menggunakan representasi aktivitas dari data training. Selanjutnya penghitungan jarak antara data masukan terhadap model data training menggunakan jarak Euclidean sebagai distance metricnya atau dalam hal ini disebut sebagai metrik *log-Euclidean*. Nilai jarak setiap log kovarian diurutkan dan diambil yang paling kecil untuk mendapatkan jarak terpendek. Jarak terpendek tersebut yang akan menunjukkan jenis aktivitas apa yang paling mirip sebagai hasil klasifikasi.

# BAB IV IMPLEMENTASI

Pembahasan implementasi ini meliputi deskripsi lingkungan tahap implementasi sistem ini dilakukan, proses-proses pada tahap implementasi yang dikerjakan, beserta penjelasan fungsi-fungsinya dalam bentuk kode sumber



## Lingkungan Pengembangan

Berikut kakas bantu yang digunakan pada proses implementasi perangkat lunak ini:

* + - 1. Windows 10 Ultimate 64 bit sebagai sistem operasi
      2. Matlab 2018a

## Implementasi Preprocessing

### *Web* *layer*

## Implementasi Koleksi Fitur Lokal

## Implementasi Representasi Aktivitas

## Implementasi Klasifikasi Aktivitas

## Uji Coba

# BAB V UJI COBA DAN EVALUASI

# BAB VI KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang dapat diambil dari hasil uji coba dan perancangan perangkat lunak sebagai jawaban dari rumusan masalah yang telah dikemukakan dan saran yang berisi pengembangan yang dapat dilakukan lebih lanjut untuk menyempurnakan perangkat lunak.



## Kesimpulan

Berikut merupakan kesimpulan yang dapat diambil dari proses pengembangan dan hasil uji coba.

## Saran

Berikut ini merupakan pengembangan lebih lanjut yang dapat dilakukan untuk menyempurnakan perangkat lunak.

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | PDDIKTI, "SRV5 PDDIKTI : Pangkalan Data Pendidikan Tinggi," PDDIKTI, 2013. [Online]. Available: http://forlap.dikti.go.id/perguruantinggi/homegraphpt. [Accessed 29 Desember 2015]. |
| [2] | A. Rouf, "PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN METODEWHITE BOX DAN BLACK BOX," *STIMIK HIMSYA JOURNAL,* vol. VIII, no. 1, pp. 3-6, 2012. |
| [3] | A. T. Averousi, Rancang Bangun Perangkat Lunak Sistem Informasi Akademik Generik Pada Modul Kurikulum, Surabaya: Institut Teknologi Sepuluh Nopember, 2015. |
| [4] | B. A. Alfirdaus, Rancang Bangun Perangkat Lunak Sistem Informasi Akademik Generik Berbasis Web Pada Modul Ekuivalensi, Surabaya: Institut Teknologi Sepuluh Nopember, 2015. |
| [5] | G. P. N. Suminto, Rancang Bangun Commercial Off The Shelf (COTS) Sistem Informasi Akademik Berbasis Web Pada Modul Kelola Pembelajaran, Surabaya: Institut Teknologi Sepuluh Nopember, 2015. |
| [6] | H. Rahman, Rancang Bangun Sistem Informasi Akademik Generik Pada Modul Penilaian Menggunakan Pola Perancangan Hierarchical Model-View-Controller, Surabaya: Institut Teknologi Sepuluh Nopember, 2015. |
| [7] | C. Walls, Modular Java - Creating Flexible Applications with OSGi and Spring, The Pragmatic Bookshelf, 2009. |
| [8] | J. Cai, R. Kapila and G. Pal, "HMVC: The layered pattern for developing strong client tiers," JavaWorld, Inc, 21 Juli 2000. [Online]. Available: http://www.javaworld.com/article/2076128/design-patterns/hmvc--the-layered-pattern-for-developing-strong-client-tiers.html. [Accessed 29 Desember 2015]. |
| [9] | "Introduction of Spring Framework," Spring, [Online]. Available: http://docs.spring.io/spring-framework/docs/3.0.x/reference/overview.html. [Accessed 6 April 2015]. |
| [10] | R. Harrop, "OSGi Concepts," Eclipse Virgo, 2011. [Online]. Available: http://www.eclipse.org/virgo/documentation/virgo-documentation-3.6.4.RELEASE/docs/virgo-user-guide/htmlsingle/virgo-user-guide.html#concepts.osgi. [Accessed 29 Desember 2015]. |
| [11] | T. A. S. Foundation, "Apache Tomcat," The Apache Software Foundation, [Online]. Available: http://tomcat.apache.org/. [Accessed 2015 Desember 29]. |
| [12] | T. P. G. D. Group, "PostgreSQL 9.3.10 Documentation," The PostgreSQL Global Development Group, [Online]. Available: http://www.postgresql.org/docs/9.3/static/intro-whatis.html. [Accessed 2015 Desember 29]. |
| [13] | J2EEBrain, "Hibernate Object Relational Mapping," J2EEBrain, [Online]. Available: http://www.j2eebrain.com/java-J2ee-hibernate-object-relational-mapping.html. [Accessed 29 Desember 2015]. |
| [14] | B. Varanasi and S. Belida, "Maven Dependency Management," in *Introducing Maven*, Apress, 2014, p. 15. |
| [15] | B. Varanasi and S. Belida, "Transitive Dependencies," in *Introducing Maven*, Apress, 2014, p. 19. |
| [16] | B. Varanasi and S. Belida, "Dependency Scope," in *Introducing Maven*, Apress, 2014, p. 21. |
| [17] | B. Varanasi and S. Belida, "Maven Life Cycle," in *Introducing Maven*, Apress, 2014, p. 37. |
| [18] | M. Fayad and D. C. Schmidt, "Object-Oriented Application Frameworks," 28 September 2006. [Online]. Available: http://www1.cse.wustl.edu/~schmidt/CACM-frameworks.html. [Accessed 2016 Januari 14]. |
| [19] | MvnRepository, "MavenRepository," Maven, 2006. [Online]. Available: http://mvnrepository.com/. [Accessed 29 Desember 2015]. |
| [20] | SpringSource, "SpringSource Enterprise Bundle Repository," SpringSource, 2008. [Online]. Available: http://ebr.springsource.com/repository/app/. [Accessed 29 Desember 2015]. |
| [21] | M. Deinum and K. Serneels, "The request processing workflow," in *Pro Spring MVC*, New York, Apress, 2012, p. 66. |

***[Halaman ini sengaja dikosongkan]***

# BIODATA PENULIS