

SISTEM INFORMASI TUGAS AKHIR & PRAKTEK KERJA LAPANGAN BERBASIS WEB MENGGUNAKAN METODE *UNIFIED PROCESS*

Ryan Adiwinata, Eko Adi Sarwoko, Indriyati

Program Studi Teknik Informatika, Fakultas MIPA, Universitas Diponegoro
punk5_minang@yahoo.co.id

ABSTRACT

Final project and job training are the compulsory subject that is organized by Informatics Department Diponegoro University. All registration process, recapitulation and administration about those subjects still have done manually. Moreover, the students and lecturers of informatics departement has not been able to access the latest informations and journals quickly and easily. The Final Project and Job Training Information System is built to solve those problems. This information system was developed using unified process method. Unified process is one of object-oriented software development process which consist of five workflows, requirement, analysis, design, implementation, and test. This information system runs on a web platform using PHP 5 scripting language and MYSQL 5 as database management system that can be accessed anytime and anywhere. The Final Project and Job Training Information System can be used as data, information and administration center about the final project and job training course.

Keywords: Final project, job training, information system, unified process, object oriented

1. Pendahuluan

Program studi Teknik Informatika Universitas Diponegoro merupakan salah satu program studi tingkat sarjana (S1) yang mewajibkan setiap mahasiswanya untuk menyelesaikan sebuah Tugas Akhir (TA) sebelum berhasil mendapatkan gelar sarjana. Selain Tugas Akhir, mahasiswa Teknik Informatika Universitas Diponegoro juga harus menyelesaikan mata kuliah Praktek Kerja Lapangan (PKL).

Seorang mahasiswa melakukan pendaftaran mata kuliah Tugas Akhir dan Praktek Kerja Lapangan dengan melampirkan beberapa persyaratan sesuai aturan yang telah ditentukan. Pendaftaran tersebut diverifikasi oleh masing – masing koordinator TA dan koordinator PKL. Namun, sampai saat ini proses pendaftaran dan proses pencatatan rekapitulasi pendaftaran masih dilakukan secara manual sehingga membutuhkan waktu yang relatif lama.

Rekapitulasi distribusi pembimbing dicatat dengan aplikasi pengolah kata *excel* atau *word*. Sehingga dibutuhkan waktu yang lama untuk membagi rata tanggungan mahasiswa yang dibimbing oleh masing – masing dosen. Oleh karena itu, koordinator TA dan koordinator PKL memerlukan suatu sistem yang secara otomatis

bisa memberikan informasi kapasitas bimbingan yang ditanggung oleh masing – masing dosen. Sistem ini diharapkan dapat mempermudah dan mengefisienkan pendistribusian dosen pembimbing.

Untuk mengatasi berbagai persoalan yang masih dihadapi oleh program studi Teknik Informatika terkait mata kuliah TA dan PKL dan juga supaya tersedianya suatu pusat informasi dan jurnal dari mata kuliah TA dan PKL serta menunjang mata kuliah tersebut berjalan dengan lancar dan efisien, program studi Teknik Informatika memerlukan suatu sistem informasi mata kuliah TA dan PKL yang dikelola sendiri oleh pihak program studi dan dapat diakses oleh seluruh *civitas akademika* program studi Teknik Informatika.

Sistem informasi TA dan PKL ini menggunakan *Unified Process* sebagai proses pengembangannya. *Unified Process* merupakan salah satu *software development process* yang menerapkan konsep berorientasi objek yang dikembangkan oleh *Ivar Jacobson*, *Grady Booch*, dan *James Rumbaugh*. [5]

Pengembangan perangkat lunak berorientasi objek menekankan konsep *reusable*. Dengan konsep *reusable*, pengembangan perangkat lunak bisa dilakukan dengan lebih cepat.

Perangkat lunak berorientasi objek lebih mudah dipelihara dan dikembangkan karena strukturnya diurai secara inheren. Sehingga lebih sedikit efek samping jika harus melakukan perubahan, dan tidak begitu menyulitkan para pengembang. Sistem berorientasi objek juga lebih mudah untuk disesuaikan. [6]

Oleh karena itu, dengan menggunakan metode *Unified Process* yang bersifat *open* dan *free* ini sehingga diharapkan akan dihasilkan perangkat lunak yang *reusable*, dan mudah untuk dipelihara.

2. Tinjauan Pustaka

2.1 Tugas Akhir

Tugas Akhir (TA) adalah kegiatan penelitian yang diusulkan, dikerjakan, dan dilaporkan oleh seorang mahasiswa secara mandiri dengan tujuan agar dapat memecahkan masalah atau menghasilkan suatu produk yang bermanfaat berdasarkan metode ilmiah yang sudah dipelajari dan dapat menuangkannya ke dalam sebuah karya ilmiah yang bisa dipertanggungjawabkan.

Mata kuliah TA merupakan mata kuliah wajib bagi mahasiswa Teknik Informatika Universitas Diponegoro sebagai syarat kelulusan sarjana. Mata kuliah ini mempunyai beban 6 SKS dan dipecah menjadi 2 bagian yaitu mata kuliah Tugas Akhir I dengan beban 2 SKS dan Tugas Akhir II dengan beban 4 SKS.

Tahapan yang harus dilalui oleh mahasiswa dalam pelaksanaan TA adalah :

a. Inisialisasi

Pada tahap ini dilakukan persiapan administratif yaitu pendaftaran mata kuliah TA serta pendistribusian dosen pembimbing.

b. Pengusulan

Kegiatan yang dilakukan pada tahap pengusulan adalah :

- Bimbingan proposal TA,
- Menghadiri beberapa seminar TA mahasiswa lain,
- Melakukan seminar proposal TA

c. Realisasi

Tahap realisasi meliputi kegiatan pelaksanaan TA, bimbingan TA dengan dosen pembimbing dan sidang TA.

d. Finalisasi

Mahasiswa mengumpulkan seluruh hasil tugas akhir yang dilaksanakan termasuk laporan dan administrasi lainnya tentang tugas akhir tersebut.

Pada program studi Teknik Informatika terdapat seorang koordinator TA yang bertugas mengatur segala administrasi yang menunjang berlangsungnya mata kuliah TA. Beberapa tugas dan fungsi yang dijalankan koordinator TA antara lain :

- a. Menerima pendaftaran TA yang dilakukan oleh mahasiswa,
- b. Mendistribusikan dosen pembimbing,
- c. Pembentukan tim penguji untuk sidang TA seorang mahasiswa,
- d. Monitoring serta evaluasi pelaksanaan pembimbingan dan pengujian TA,
- e. Membuat laporan administrasi TA.

2.2 Praktek Kerja Lapangan

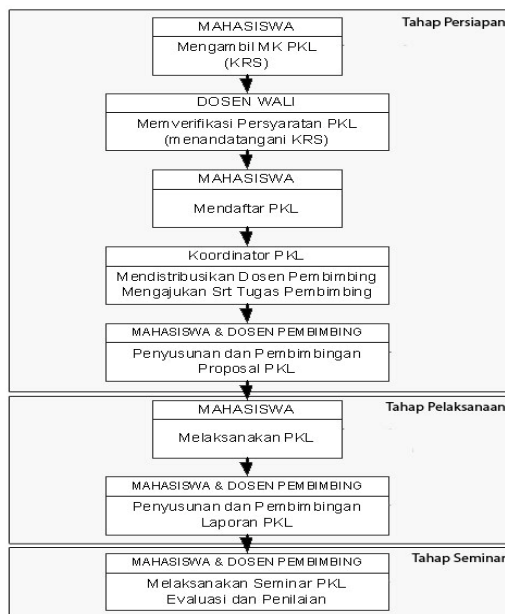
Praktek Kerja Lapangan (PKL) merupakan mata kuliah yang wajib diambil oleh mahasiswa Teknik Informatika Universitas Diponegoro. Mata kuliah yang terdiri atas 2 sks ini bertujuan untuk memberi bekal kepada mahasiswa agar mengetahui permasalahan di dunia kerja yang berkaitan dengan mata kuliah yang pernah diambil sebelumnya dan dapat membuat deskripsi teoritis dan analisis dari sudut pandang bidang studi yang bersangkutan serta memberikan / menjelaskan alternatif pemecahannya dalam bentuk laporan untuk diseminarkan.

Pelaksanaan mata kuliah PKL ini bisa dikelompokkan ke dalam tiga tahapan, yaitu : tahap persiapan, tahap pelaksanaan dan tahap seminar.

Pada tahap persiapan, mahasiswa harus mengambil mata kuliah PKL melalui KRS yang sudah diverifikasi oleh dosen wali. Proses selanjutnya adalah mahasiswa mendaftar kepada koordinator PKL dengan memenuhi beberapa persyaratan yang telah ditetapkan. Koordinator PKL akan memverifikasi pendaftaran, mendistribusikan dosen pembimbing dan memberikan surat tugas pembimbingan PKL kepada dosen pembimbing mahasiswa tersebut.

Pada tahap persiapan ini juga berlangsung proses penyusunan dan pembimbingan proposal PKL.

Tahap pelaksanaan dimulai dari hari pertama mahasiswa melaksanakan PKL di tempat PKL masing – masing sampai dengan penyusunan dan pembimbingan laporan PKL. Setelah penyusunan laporan selesai dilaksanakan kemudian barulah dilaksanakan seminar PKL dan evaluasi atau penilaian. Tahapan pelaksanaan mata kuliah PKL bisa dilihat pada gambar 1



Gambar 1 Tahapan Pelaksanaan Mata Kuliah PKL

2.3 Unified Process

Unified Process merupakan salah satu software development process. Software development process adalah sekumpulan aktivitas yang dibutuhkan untuk mentransformasikan permintaan user ke dalam sistem perangkat lunak. Unified Process merupakan sebuah framework yang bisa dispesialisasikan berdasarkan banyak kelompok, perbedaan tingkat kemampuan, dan perbedaan skala proyek. Unified Process bersifat component-based, artinya sistem perangkat lunak yang dibangun dibuat dari sejumlah

komponen yang diinterkoneksi melalui interface yang dibuat dengan baik. [5]

Unified Process menggunakan Unified Modeling Language (UML) untuk mempersiapkan seluruh blueprint dari sistem perangkat lunak. Unified Process dan UML dikembangkan dengan saling berkesinambungan.

Unified Process mempunyai tiga karakteristik utama, yaitu:

a. Use Case Driven

Use case merupakan sebuah potongan dari keseluruhan fungsionalitas yang dilakukan oleh sistem. Use case bukan hanya merupakan tool untuk menggambarkan kebutuhan dari sistem, tetapi use case juga mengarahkan semua kebutuhan tersebut menuju proses desain, implementasi dan pengujian. [5]

b. Architecture Centric

Arsitektur merupakan gambaran mengenai keseluruhan desain yang menonjolkan beberapa karakteristik. Arsitektur dengan use case haruslah seimbang karena sebuah use case apabila direalisasikan harus disesuaikan dengan arsitektur yang dikembangkan dan juga sebaliknya arsitektur juga harus memberikan ruang untuk realisasi dari semua use case yang diminta, baik pada awal pengembangan hingga langkah selanjutnya. Architecture centric berarti suatu arsitektur sistem dipergunakan sebagai satu artifak utama untuk mengkonsep, membangun dan pengelolaan sistem perangkat lunak.

c. Iterative & Incremental

Iterative mengacu pada semua tahapan pada workflow dilalui secara berulang (iterasi) hingga sebuah use case berhasil terealisasi. Dalam setiap iterasi, dilakukan identifikasi dan penetapan use case yang relevan, membuat sebuah desain menggunakan arsitektur terpilih sebagai sebuah panduan, mengimplementasikan desain tersebut ke dalam komponen – komponen dan kemudian memverifikasi komponen tersebut. Setiap rilis produk selalu ada review agar pengembangan selanjutnya bisa lebih baik.

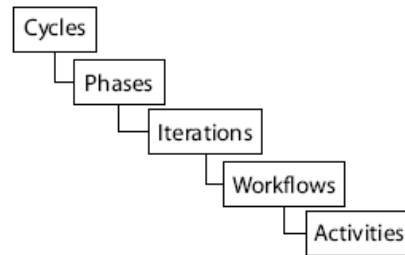
Review ini juga bertujuan untuk mengganti prosedur yang mungkin kurang sesuai.

Pada *Unified Process*, pengembangan perangkat lunak dilakukan dalam satu siklus atau lebih dari satu siklus yang berurutan. Suatu siklus akan berakhir dengan *release* sebuah produk (contoh : versi 1.0, versi 1.1). Dalam satu siklus terbagi atas empat fase. Fase tersebut adalah fase *inception*, *elaboration*, *construction*, dan *transition*. Didalam suatu fase bisa dilakukan beberapa *iteration* atau perulangan sejumlah *workflow*. *Workflow* tersebut yaitu *requirement*, *analysis*, *design*, *implementation*, dan *test*.

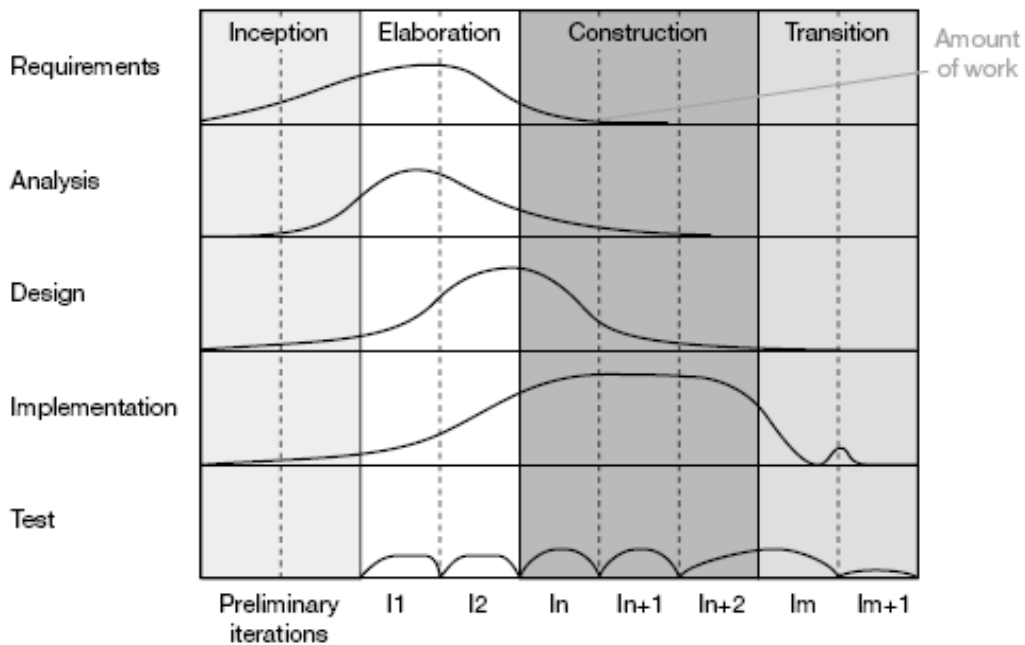
Pada masing – masing *workflow* dilakukan sejumlah *activities*. Didalam satu *activities* dihasilkan artifak – artifak yang diperlukan.

Struktur hirarki *Unified Process* ditunjukkan pada gambar 2.2 [4]

Saat *project* beralih dari satu fase ke fase berikutnya, jumlah kerja yang dilakukan pada setiap *workflow* juga berubah. Gambar 3 menunjukkan hubungan antara fase dengan *workflow*. [1]



Gambar 2 Struktur Hirarki *Unified Process*



Gambar 3 Hubungan Fase dengan *Workflows* dalam *Unified Process*

Penjelasan dari masing – masing fase yang ada dalam *Unified Process* adalah sebagai berikut :

a. Inception

Penekanan utama dari fase ini adalah pada *workflow requirement* dan sedikit *analysis*.

Selama fase *inception*, sebuah ide atau asumsi dikembangkan untuk mengetahui permintaan tentang batasan dari perangkat lunak.

b. *Elaboration*

Fase *elaboration* berfokus pada *workflow analysis*. Pada fase ini, pekerjaan yang dilakukan yaitu melengkapi *requirement*, baik fungsional maupun non-fungsional serta melakukan analisis dan pembuatan desain terhadap apa yang akan dibangun. Salah satu aktivitas utama dalam fase ini adalah membuat arsitektur sistem. Pada fase ini, juga telah dilakukan sedikit implementasi dan pengujian untuk mengeksekusi arsitektur yang diciptakan tersebut.

c. *Construction*

Pada fase ini sebuah rancangan produk mulai direalisasikan. Tujuan dari fase *construction* adalah melengkapi seluruh *workflow requirement, analysis* dan *design* serta membangun arsitektur yang akan ditransformasikan ke dalam sistem akhir. Pada akhir fase ini seluruh *use case* sudah dipenuhi dan produk akan diluncurkan sesuai dengan persetujuan *customer*.

d. *Transition*

Menempatkan sistem pada lingkungan status *beta release*. Pada *beta release* inisejumlah *user* akan mencoba produk yang telah dihasilkan dan melaporkan cacat atau defisiensi yang ditemukan. Kemudian pengembang melakukan koreksi atas masalah yang dilaporkan untuk *general release* yang akan dihadirkan pada kumpulan *user* yang lebih luas lagi. Fase ini berakhir dengan *release* produk resmi.

Aktivitas yang dilakukan dalam kelima *core workflow* adalah sebagai berikut :

a. *Requirement*

Sebagian besar pekerjaan pada *requirement* ini terjadi dalam fase *inception* dan *elaboration*. *Requirement* merupakan *workflow* yang sulit karena harus mengetahui apa yang dibutuhkan oleh *user* dan terkadang *user* tidak begitu mengetahui bagian pekerjaan mereka yang akan dipermudah atau digantikan oleh sistem.

Ada dua tipe *requirement* yaitu *requirement* fungsional dan non-fungsional. *Requirement* dari sistem direpresentasikan melalui *use*

case, kemudian pada akhirnya akan menghasilkan sebuah *use case model* yang terdiri atas *use case*, aktor, dan artefak.

b. *Analysis*

Analisis bertujuan untuk menganalisa *requirement* yang telah diidentifikasi melalui *use case* dan menyusunnya ke dalam lingkup internal sistem. [3]

Akan tetapi pada kenyataannya juga dilakukan sedikit perbaikan terhadap *requirement* yang salah atau menyimpang. Pada *workflow* analisis ini dihasilkan suatu model analisis yang didapat dari mengidentifikasi elemen / entitas utama dari sistem yang diperlukan untuk memenuhi *userrequirement*.

Beberapa artefak yang dihasilkan dalam *workflow analysis* antara lain (Jacobson, Et all 1999) :

- *Analysis class*
- *Use case realization – analysis*
- *Analysis package*

c. *Design*

Pada alur kerja atau *workflow design*, kegiatan yang dilakukan ialah menggambarkan sistem (termasuk arsitektur sistem) yang memenuhi semua *requirement* termasuk seluruh *non-functional requirement* dan batasan – batasan lainnya. Input yang paling utama pada *design* yaitu hasil dari *analysis* tepatnya model analisis. Dengan menggunakan model analisis, dilakukan spesifikasi secara penuh bagaimana fungsionalitas akan diimplementasikan.

d. *Implementation*

Implementation adalah mentransformasikan sebuah *design model* ke dalam *source code, script, binaries, executable* dan lainnya. *Workflow implementation* lebih berfokus pada fase *construction*, seperti yang ditunjukkan oleh gambar 2.3.

e. *Test*

Untuk memastikan bahwa sistem dapat menyediakan fungsionalitas yang diperlukan maka dilakukan pengujian. Pada *workflow test*, dilakukan pengujian pada setiap bagian dari yang dihasilkan oleh *workflow*

implementaion. Pengujian ini dapat dilakukan dengan membuat *test model*, *test plan*, *test case*, dan lain sebagainya.

2.4 Unified Modelling Language

Unified Modelling Language (UML) adalah keluarga notasi grafis yang membantu pendeskripsian dan desain perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. UML merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG). OMG adalah sebuah konsorsium terbuka yang dibentuk untuk membuat dan mengeluarkan standar – standar teknologi *object oriented* (<http://www.omg.org>).

Pengembang perangkat lunak menggunakan UML untuk membuat sketsa atau *blueprint* dari sebuah sistem yang meliputi konsep *process business*, penulisan *class* dalam bahasa pemrograman yang spesifik, skema basis data serta komponen – komponen yang diperlukan oleh sistem. UML versi terbaru adalah UML versi 2.0.

UML terdiri atas tiga elemen utama penyusunnya, yaitu : *thing*, *relationship* dan *diagram*.

a. Things

Things atau benda adalah hal yang mendasar dalam model UML karena merupakan bagian paling statik dari sebuah model, serta menjelaskan elemen – elemen lainnya dari sebuah konsep. *Things* bisa dibagi ke dalam empat kategori, yaitu :

- *Structural*

Structural things adalah bagian statis dari model UML yang merepresentasikan kata benda (*noun*). Contohnya : *class*, *interface*, *collaboration*, *use case*, *active class*, *component*, *artifact*, *node*.

- *Behavioral*

Behavioral things merupakan bagian dinamis (*verbs*) dari model UML yang merepresentasikan perilaku (*behavior*). Jenis *behavioral things* seperti *interactions* dan *state machines*.

- *Grouping*

Grouping adalah elemen berbentuk *packages* yang digunakan untuk mengelompokkan elemen model yang saling berhubungan secara semantik menjadi sebuah unit.

- *Annotational*

Annotational adalah elemen berbentuk *notes* yang bisa ditambahkan ke dalam sebuah model sebagai informasi khusus pada model tersebut.

b. Relationships

Relationship atau relasi memberikan gambaran bagaimana hubungan satu sama lain antara dua atau lebih benda (*things*). Ada beberapa tipe relasi dalam UML, yaitu :

- *Association*, merupakan deskripsi dari sekumpulan *link* antara objek.

- *Dependency*, sebuah perubahan pada satu benda yang memiliki efek terhadap benda lain yang bergantung pada benda tersebut.

- *Generalization*, satu elemen menjadi spesialisasi dari yang lain dan boleh digantikan oleh elemen umum yang lain.

- *Realization*, sebuah relasi antara *classifiers*.

c. Diagram

Diagram menunjukkan sekumpulan dari benda yang menjelaskan sebuah cerita tentang sistem perangkat lunak dan merupakan salah satu cara untuk memvisualisasikan apa yang akan dikerjakan oleh sistem dan bagaimana sistem melakukannya. UML versi 2.0 terdiri atas 13 jenis diagram resmi, seperti yang dijelaskan pada tabel 1. [2]

Tabel 1 Nama – Nama Diagram

Nama Diagram	Kegunaan
<i>Activity</i>	<i>Behavioral prosedural</i> dan parallel
<i>Class</i>	<i>Class</i> , fitur, dan hubungan – hubungannya
<i>Communication</i>	Interaksi antar objek, penekanan pada jalur
<i>Component</i>	Struktur dan koneksi komponen
<i>Composite structure</i>	Dekomposisi <i>runtime</i> sebuah <i>class</i>
<i>Deployment</i>	Pemindahan artifak ke <i>node</i>
<i>Interaction overview</i>	Campuran <i>sequence</i> dan <i>activity diagram</i>
<i>Object</i>	Contoh konfigurasi dari contoh – contoh
<i>Package</i>	Struktur hirarki <i>compile-time</i>
<i>Sequence</i>	Interaksi antar objek, penekanan pada <i>sequence</i>
<i>State machine</i>	Bagaimana <i>event</i> mengubah objek selama aktif
<i>Timing</i>	Interaksi antar objek, penekanan pada <i>timing</i>
<i>Use Case</i>	Bagaimana pengguna berinteraksi

3. Definisi Kebutuhan, Analisis, dan Perancangan

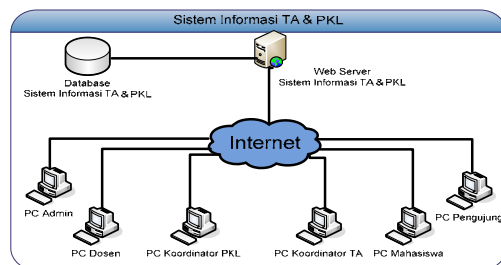
3.1 Definisi Kebutuhan

Sistem informasi ini dikembangkan dengan menggunakan *platformweb* dan bisa diakses melalui jaringan yang terkoneksi ataupun internet. Terdapat enam jenis otoritas pengguna yaitu sebagai pengunjung, mahasiswa, dosen, koordinator PKL, koordinator TA, dan administrator. Fungsi utama dari perangkat lunak ini antara lain :

- a. Menerima pendaftaran mata kuliah TA dan PKL,

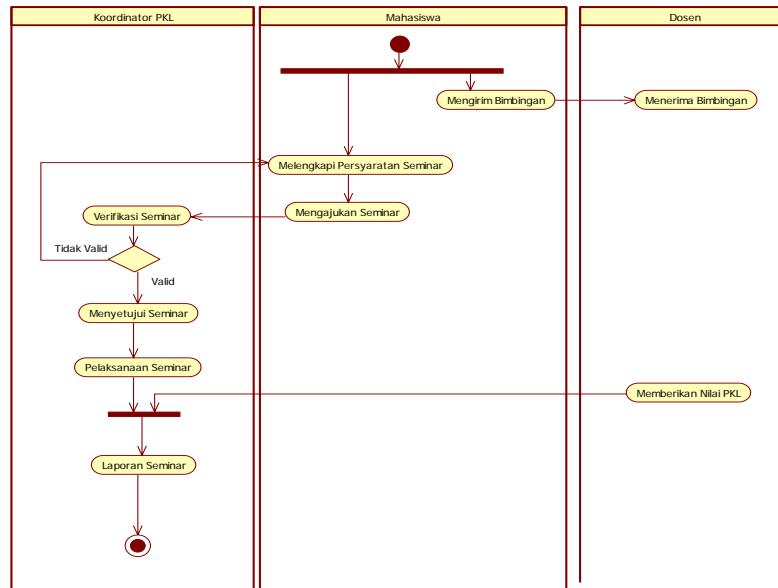
- b. Melakukan pendistribusian dosen pembimbing TA dan PKL,
- c. Menyediakan fasilitas untuk mendokumentasikan riwayat bimbingan TA dan PKL,
- d. Menyediakan fasilitas untuk mengajukan jadwal seminar PKL dan seminar atau sidang TA kepada masing – masing koordinator,
- e. Sebagai pusat informasi dan jurnal mata kuliah TA dan PKL,
- f. Menyediakan berbagai dokumen dan laporan tentang mata kuliah TA dan PKL yang dibutuhkan oleh seluruh *civitas akademika* program studi Teknik Informatika, terutama untuk Koordinator PKL dan Koordinator TA.

Perangkat lunak ini dibangun dengan menggunakan *software open source*. Secara garis besar arsitektur dan deskripsi umum Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan ini ditunjukkan oleh gambar 4.

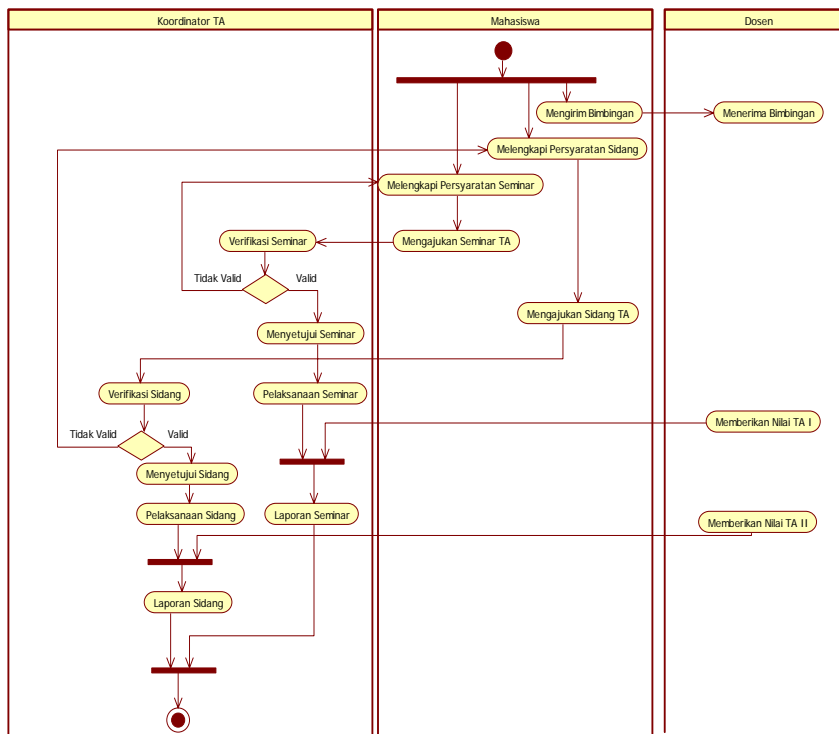


Gambar 4 Deskripsi Umum Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan

Activity diagram merupakan suatu teknik atau pemodelan untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. *Activity diagram* disebut juga dengan *OO flowchart*, artinya diagram alir dalam konsep berorientasi objek. *Activity diagram* pada Sistem Informasi Tugas Akhir & Praktek Kerja Lapangan bisa dilihat pada gambar 5.



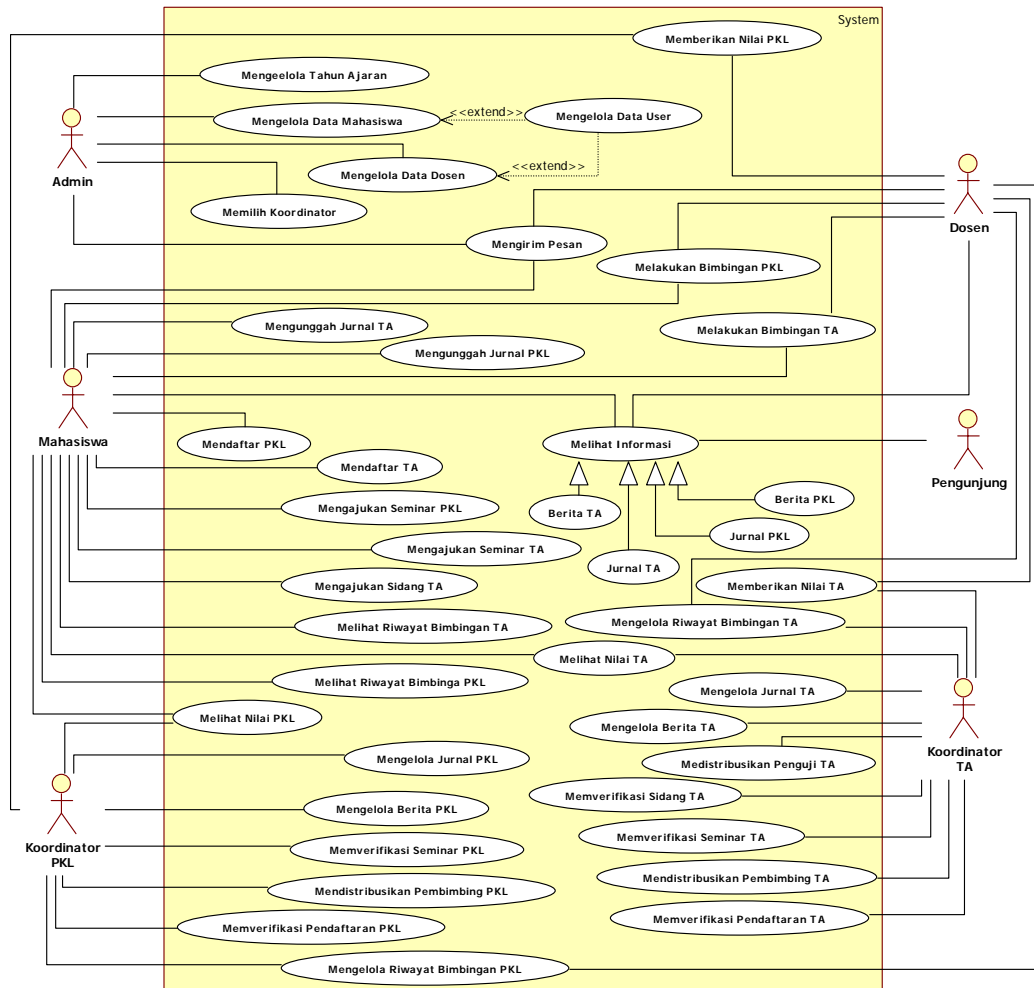
Gambar 5 Activity Diagram dari Proses PKL



Gambar 6 Activity Diagram dari Proses TA

System use case diagram terbentuk dari serangkaian aktor dan use case. System use case diagram dari Sistem Informasi Tugas Akhir dan

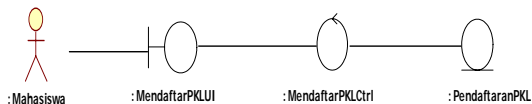
Praktek Kerja Lapangan dapat dilihat pada gambar 7.



Gambar 7 System Use Case Diagram

3.2 Analisis

Hasil analisis dari pengembangan sistem informasi ini meliputi *use case realization - analysis* dan *analysis class. Use case realization - analysis* menunjukkan interaksi antara *analysis class* dengan fungsional perangkat lunak.



Gambar 8 Analysis Class Diagram Mendafar PKL

Realisasi *use case* Mendafar PKL ditunjukkan *analysis class model* pada Gambar

8. Analisis class model Mendafar PKL menunjukkan *class-class* yang terkait pada *use case* Mendafar PKL, yaitu MendafarPKLUI (*boundary class*), MendafarPKLCtrl (*control class*), dan PendaftaranPKL (*entity class*).

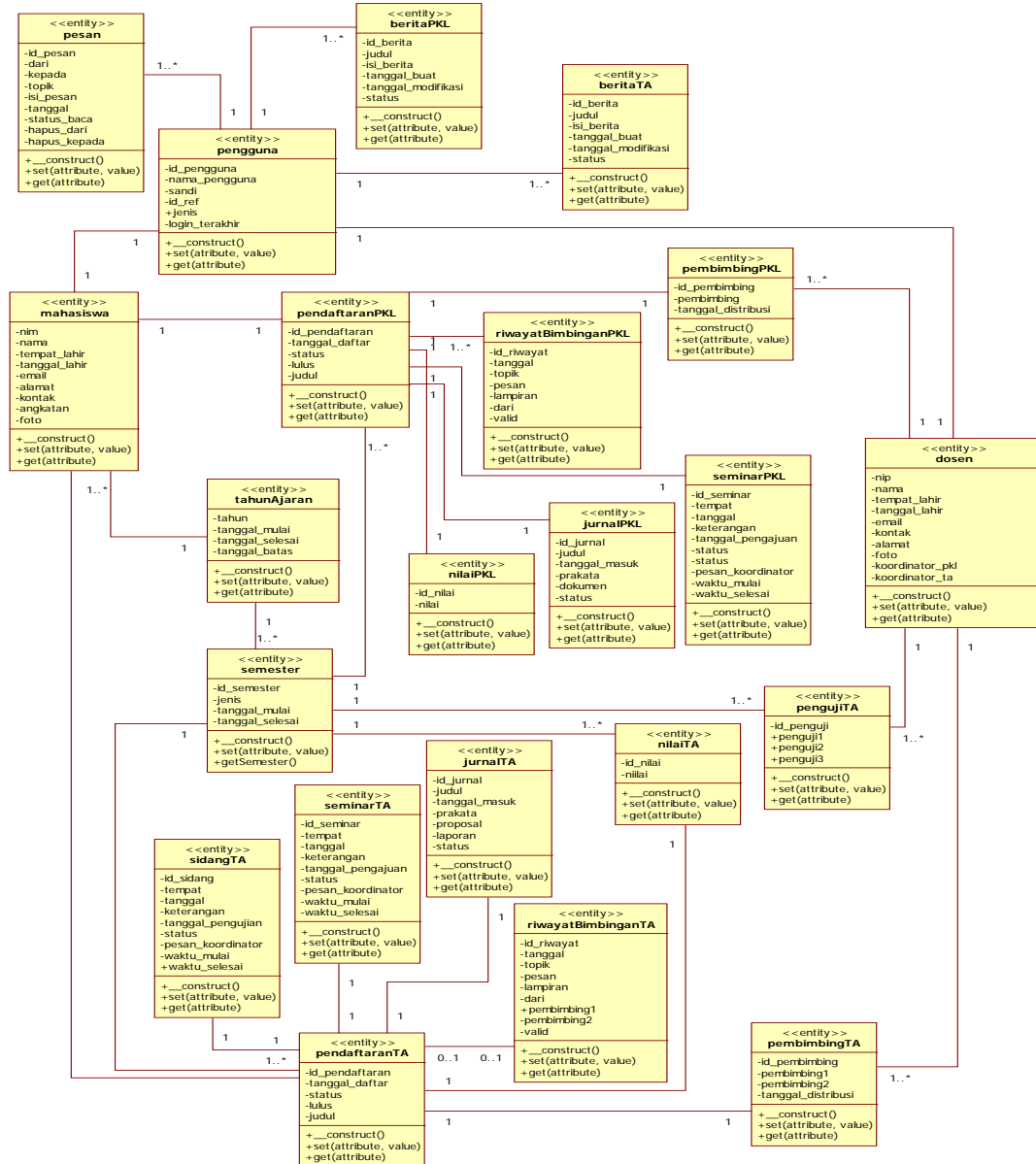
3.3 Perancangan

Pada tahap perancangan dihasilkan realisasi *use case* yang dapat digambarkan dengan menggunakan *class diagram, sequence diagram*. Berikut contoh realisasi *use case* pada tahap perancangan untuk *use case* MendafarPKL.

Tabel 2 Identifikasi Class Perancangan Use Case Mendafar PKL

Stereotype	Class Analisis	Class Perancangan
Boundary	MendaftarPKLUI	PklUI
Control	MendaftarPKLCtrl	PklCtrl
Entity	PendaftaranPKL	PendaftaranPKL

Selain itu, pada tahap perancangan juga dihasilkan *class diagram* dan perancangan *database*. Berikut adalah *class diagram* dan perancangan *database* dari sistem informasi ini.



Gambar 9 Class Diagram Class Entity

4. Implementasi dan Pengujian

4.1 Implementasi

Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan diimplementasikan menggunakan web server Apache 2.2.11, DBMSMySQL versi 5.0.51a, dan bahasa pemrograman PHP versi 5.2.8.

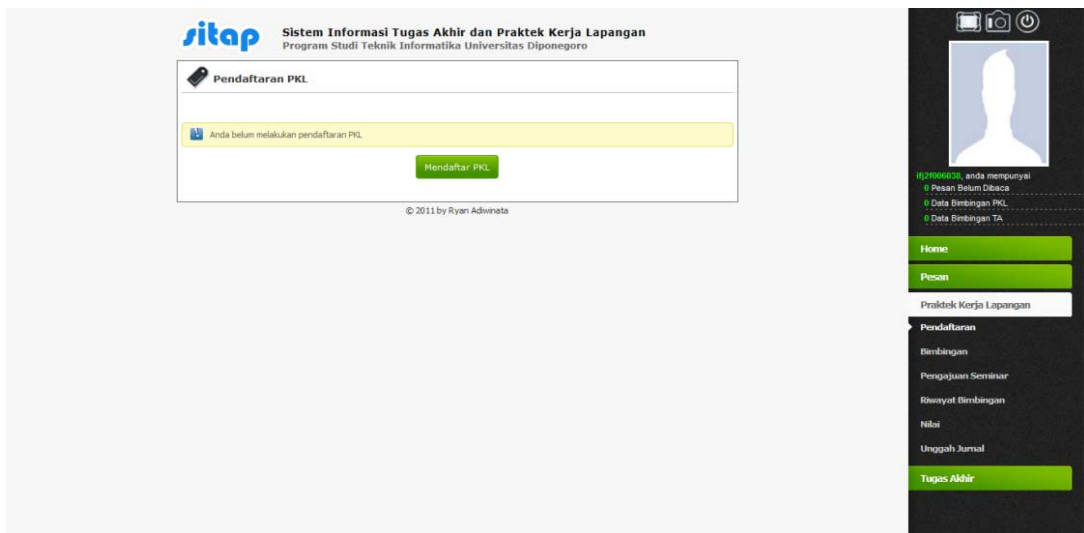
a. Implementasi Class

Sistem informasi Tugas Akhir dan Praktek Kerja Lapangan ini diimplementasi dengan menggunakan *framework php* yang dikembangkan sendiri. *Framework* tersebut

menggunakan konsep MVC (*Model View Controller*).

b. Implementasi Antarmuka

Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan ini mempunyai dua antarmuka utama, yaitu : antarmuka untuk pengunjung yang bisa melihat informasi dan referensi jurnal dari mata kuliah Tugas Akhir dan Praktek Kerja Lapangan dan antarmuka untuk pengguna yaitu mahasiswa dan dosen. Gambar 10 merupakan hasil implementasi antarmuka untuk mahasiswa dan dosen.



Gambar 10 Implementasi Antarmuka Mahasiswa dan Dosen

4.2 Pengujian

Pengujian perangkat lunak Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan dilakukan dengan metode *black box*, yaitu menguji fungsionalitas dari perangkat lunak, tanpa harus mengetahui struktur internal program. Dari hasil pengujian, dapat diketahui bahwa Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan ini telah memenuhi untuk :

- Melakukan pendaftaran mata kuliah Tugas Akhir dan Praktek Kerja Lapangan.
- Memverifikasi pendaftaran.
- Melakukan rekapitulan pendaftaran dan distribusi pembimbing mata kuliah Tugas

Akhir dan Praktek Kerja Lapangan tiap semesternya.

- Mahasiswa dan dosen bisa saling berinteraksi dalam melakukan bimbingan Tugas Akhir atau Praktek Kerja Lapangan.
- Mahasiswa bisa mengajukan jadwal seminar Praktek Kerja Lapangan, seminar dan sidang Tugas Akhir.
- Dosen pembimbing memberikan nilai Tugas Akhir dan Praktek Kerja Lapangan.
- Menampilkan informasi dan jurnal terkini dari mata kuliah Tugas Akhir dan Praktek Kerja Lapangan.

5. Kesimpulan dan Saran

Kesimpulan yang dapat diambil dalam penelitian ini adalah dihasilkan sebuah Sistem

Informasi Tugas Akhir dan Praktek Kerja Lapangan yang dapat membantu proses mata kuliah Tugas Akhir dan Praktek Kerja Lapangan di lingkungan program studi Teknik Informatika Universitas Diponegoro bisa berjalan dengan lancar. Selain itu, dengan adanya sistem informasi ini diharapkan bisa membantu pekerjaan masing – masing koordinator TA dan PKL dalam melakukan administrasi mata kuliah tersebut. Mahasiswa dan dosen program studi Teknik Informatika juga bisa memanfaatkan sistem informasi ini sebagai sumber informasi dan data jurnal tentang mata kuliah TA dan PKL.

Sistem Informasi Tugas Akhir dan Praktek Kerja Lapangan ini masih memerlukan pengembangan lebih lanjut. Salah satu hal yang harus dikembangkan yaitu penerapan basis data berorientasi objek. Hal ini bertujuan agar basis data lebih sesuai dengan metode pengembangan dari aplikasi ini. Selain itu, sangat diharapkan peran serta *civitas akademika* untuk bisa memanfaatkan sistem informasi ini agar bisa

menunjang proses mata kuliah Tugas Akhir dan Praktek Kerja Lapangan berjalan dengan efisien.

6. Daftar Pusaka

- [1] Arlow J. dan Neustadt I., 2002, *UML and The Unified Process Practical Object-Oriented Analysis & Design*, Addison Wesley, USA.
- [2] Fowler, Martin, 2005, *UML Distilled Edisi 3*. (Andi, Yogyakarta)
- [3] Ian Sommerville.2003. *Software Engineering (Rekayasa Perangkat Lunak)*. (Erlangga, Jakarta)
- [4] Hunt J., 2003, *Guide to the Unified Process featuring UML, Java and Design Patterns*, Springer, London
- [5] Jacobson, Ivar, Grady Booch, dan James Rumbaugh, 1999, *The Unified Software Development Process*. Addison Wesley.
- [6] Pressman, Roger S., 2001, *Software Engineering : A Practitioner's Approach Fifth Edition*. (McGraw – Hill, New York)