

**PENGEMBANGAN SISTEM INFORMASI
SELEKSI BEASISWA UNGGULAN P3SWOT KEMDIKNAS ONLINE
MENGUNAKAN UNIFIED PROCESS**



SKRIPSI

Disusun oleh:

Khadijah

J2F 006 026

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO**

2011

**PENGEMBANGAN SISTEM INFORMASI
SELEKSI BEASISWA UNGGULAN P3SWOT KEMDIKNAS ONLINE
MENGUNAKAN UNIFIED PROCESS**



SKRIPSI

**Disusun Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Komputer**

**Disusun oleh:
Khadijah
J2F 006 026**

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS DIPONEGORO**

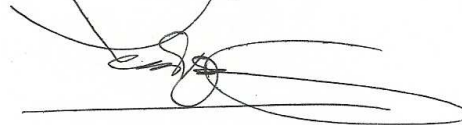
2011

HALAMAN PENGESAHAN

Judul : Pengembangan Sistem Informasi Seleksi Beasiswa Unggulan P3SWOT
Kemdiknas *Online* Menggunakan *Unified Process*
Nama : Khadijah
NIM : J2F 006 026

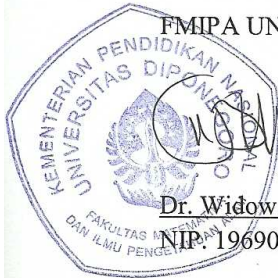
Telah diujikan pada sidang tugas akhir pada tanggal 8 April 2011 dan dinyatakan lulus pada tanggal 18 April 2011.

Semarang, 18 April 2011
Panitia Penguji Tugas Akhir
Ketua,



Drs. Eko Adi Sarwoko, M.Kom
NIP.19651107 199203 1 003

Mengetahui,
Ketua Jurusan Matematika
FMIPA UNDIP



Dr. Widowati, S.Si, M.Si
NIP. 19690214 199403 2 002

Mengetahui,
Sekretaris Program Studi Teknik Informatika
Jurusan Matematika FMIPA UNDIP



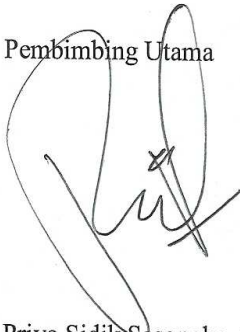
Aris Sugiharto, S.Si., M.Kom
NIP. 19710811 199702 1 004

HALAMAN PENGESAHAN

Judul : Pengembangan Sistem Informasi Seleksi Beasiswa Unggulan P3SWOT
Kemdiknas *Online* Menggunakan *Unified Process*
Nama : Khadijah
NIM : J2F 006 026

Telah diujikan pada sidang tugas akhir pada tanggal 8 April 2011.

Pembimbing Utama



Priyo Sidik Sasongko, M.Kom
NIP. 19700705 199702 1 001

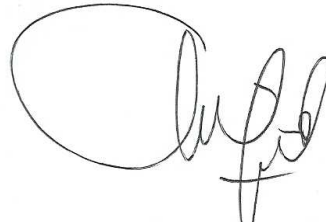
Pembimbing Anggota



Edy Suharto, ST
NIP. 19800914 200604 1 002

Semarang, 18 April 2011

Pembimbing Anggota



Dr. A.B. Susanto, M.Sc
NIP. 19640501 198902 1 001

ABSTRAK

Kementerian Pendidikan Nasional (Kemdiknas) telah mencanangkan program beasiswa unggulan, salah satunya adalah beasiswa unggulan P3SWOT (Peneliti, Penulis, Pencipta, Seniman, Wartawan, Olahragawan, dan Tokoh). Sebagai penyelenggara program beasiswa unggulan P3SWOT, Sekretariat Beasiswa Unggulan BPKLN Kemdiknas telah menerapkan pemanfaatan teknologi informasi dalam bentuk *website* yang menyediakan informasi pendaftaran, formulir pendaftaran yang dapat di-*download* oleh calon pendaftar, dan pengumuman pemenang beasiswa. Namun, *website* yang dikembangkan cenderung bersifat statis. Mekanisme seleksi beasiswa unggulan P3SWOT yang berjalan saat ini juga tidak terlepas dari sejumlah persoalan, yaitu proses pendaftaran yang kurang efektif dan tidak terintegrasinya data pendaftaran beasiswa. Sistem informasi Seleksi Beasiswa Unggulan P3SWOT Kemdiknas *Online* dibangun untuk mengatasi permasalahan tersebut. Sistem informasi ini dikembangkan menggunakan *Unified Process*. *Unified Process* merupakan salah satu *object oriented software development process* yang terdiri atas 5 *workflows*, yaitu *requirement*, *analysis*, *design*, *implementation*, dan *test*. Sistem informasi ini dibangun menggunakan bahasa pemrograman PHP 5 dan sistem manajemen basis data MySQL 5. Sistem informasi ini dapat digunakan untuk menangani proses seleksi beasiswa unggulan P3SWOT meliputi proses pendaftaran, verifikasi dan penilaian pendaftaran, penetapan pemenang, dan pelaporan.

Kata kunci : sistem informasi, beasiswa unggulan P3SWOT, *Unified Process*, berorientasi objek.

ABSTRACT

The Ministry of National Education has declared excellence scholarship program, one kind of these excellence scholarships is P3SWOT. P3SWOT is an excellence scholarship given to researcher, writer, creator, artist, journalist, athlete, and figure. As a holder of P3SWOT excellence scholarship, the Secretary of Excellence Scholarship, Bureau Planning and International Cooperation, the Ministry of National Education has been taking the advantage of the information technology by building a website that provides registration information, a registration form that can be downloaded by registrant, and the announcement of the scholarship winners. However, the website is not optimal because it is static. The selection mechanism of P3SWOT excellence scholarship still has some problems. Recently there are ineffective registration and unintegrated scholarship registration data. The online selection of P3SWOT excellence scholarship information system was developed to solve those problems. This information system was developed using Unified Process. Unified Process is an object oriented software development process that consists of five workflows. They are requirement, analysis, design, implementation, and test. This information system was built using PHP 5 programming language and MySQL 5 database management system. This information system can be used to handle the selection of P3SWOT excellence scholarship including registration, registration verification and scoring, deciding the winners, and reporting.

Keywords : information system, P3SWOT excellence scholarship, Unified Process, object oriented.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan tugas akhir yang berjudul **“Pengembangan Sistem Informasi Seleksi Beasiswa Unggulan P3SWOT Kemdiknas Online Menggunakan *Unified Process*”** dengan baik dan lancar. Laporan tugas akhir ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) pada Program Studi Teknik Informatika Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro Semarang.

Sebagai pelaksanaan penyusunan laporan tugas akhir ini, penulis banyak mendapat bimbingan, arahan, dan bantuan dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati, penulis ingin mengucapkan terima kasih dengan tulus kepada :

1. Dr. Muhammad Nur, DEA, selaku Dekan FMIPA UNDIP.
2. Dr. Widowati, S.Si, M.Si, selaku Ketua Jurusan Matematika FMIPA UNDIP.
3. Drs. Eko Adi Sarwoko, M.Kom, selaku Ketua Program Studi Teknik Informatika FMIPA UNDIP.
4. Nurdin Bahtiar, S.Si., M.T, selaku Koordinator Tugas Akhir Program Studi Teknik Informatika FMIPA UNDIP.
5. Priyo Sidik Sasongko, S.Si., M.Kom, selaku dosen pembimbing I yang telah membimbing dan mengarahkan penulis dalam menyelesaikan tugas akhir ini.
6. Edy Suharto, S.T, selaku dosen pembimbing II yang telah membimbing dan mengarahkan penulis dalam menyelesaikan tugas akhir ini.
7. Dr. A.B. Susanto, M.Sc, selaku dosen pembimbing III dan Koordinator Beasiswa Unggulan BPKLN Kemdiknas atas kesempatan, bimbingan, dan bantuan yang diberikan dalam mengerjakan tugas akhir ini.
8. Retno Kusumaningrum, S.Si, M.Kom, dan Drs. Kushartantya, MI.Komp, selaku dosen wali yang memberikan arahan dalam bidang akademik.
9. Bapak dan Ibu dosen Program Studi Teknik Informatika FMIPA UNDIP atas semua ilmu yang telah diberikan.
10. Seluruh Staf Sekretariat Beasiswa Unggulan BPKLN (Biro Perencanaan dan Kerjasama Luar Negeri) Kementerian Pendidikan Nasional atas kesempatan, saran, masukan, dan bantuan yang diberikan dalam mengerjakan tugas akhir ini.

11. Orang tuaku tercinta, kakak, dan adik-adikku, yang telah memberikan dukungan moril, material, perhatian, do'a, dan kasih sayangnya selalu.
12. Nani, Aisyah, Lia, dan Kaka yang telah banyak berbagi serta memberikan bantuan, masukan, dan motivasi.
13. Mba Anita atas kesempatan, dukungan, dan bantuannya, Suryo atas bantuan *design*-nya, dan teman-teman mahasiswa Teknik Informatika UNDIP 2006 (eSPeGe), kakak dan adik angkatan atas bantuan dan dukungannya.
14. Semua pihak yang telah membantu hingga selesainya tugas akhir ini, yang tidak dapat penulis sebutkan satu persatu. Semoga Allah membalas segala kebaikan yang telah diberikan kepada penulis.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan tugas akhir ini, untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun demi kesempurnaan tugas akhir ini. Semoga laporan tugas akhir ini dapat bermanfaat bagi pengembangan ilmu dan pengetahuan, khususnya pada bidang teknologi informasi.

Semarang, April 2011

Penulis

DAFTAR ISI

	Hal
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB I PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Rumusan Masalah	2
I.3. Tujuan dan Manfaat	2
I.4. Ruang Lingkup	3
I.5. Sistematika Penulisan	3
BAB II DASAR TEORI	5
II.1. Sistem Informasi	5
II.2. Beasiswa Unggulan P3SWOT Kemdiknas	5
II.3. Konsep <i>Object Oriented</i>	7
II.4. <i>Unified Modeling Language</i>	8
II.5. <i>Unified Process</i>	13
II.6. Bahasa Pemrograman PHP <i>Hypertext Presprocessor</i>	19
II.7. <i>Framework CodeIgniter</i>	20
II.8. Sistem Manajemen Basis Data MySQL	21

DAFTAR GAMBAR

	Hal
Gambar 2.1 Contoh <i>Class Diagram</i>	11
Gambar 2.2 Simbol <i>Use Case</i>	11
Gambar 2.3 Simbol Aktor.....	11
Gambar 2.4 Contoh <i>Sequence Diagram</i>	12
Gambar 2.5 Hubungan Fase dengan <i>Workflow</i> dalam <i>Unified Process</i>	14
Gambar 2.6 <i>Flow Chart</i> Aplikasi dalam CodeIgniter	20

DAFTAR TABEL

	Hal
Tabel 2.1 Jenis <i>Relationship</i> pada <i>Class Diagram</i>	10
Tabel 2.2 Jenis <i>Relationship</i> pada <i>Use Case</i>	11
Tabel 2.3 Notasi <i>Business Use Case Model</i>	16
Tabel 2.4 Jenis <i>Analysis Class</i>	17

BAB I

PENDAHULUAN

Pada bab ini diuraikan latar belakang, rumusan permasalahan, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan dalam pembuatan tugas akhir ini.

I.1. Latar Belakang

Program beasiswa unggulan telah dicanangkan oleh Kementerian Pendidikan Nasional (Kemdiknas) dalam rangka meningkatkan daya saing bangsa Indonesia di segala bidang, khususnya mutu pendidikan dan pengembangan potensi sumber daya daerah yang dinilai masih sangat rendah. Salah satu jenis beasiswa unggulan yang ditawarkan oleh Kemdiknas adalah beasiswa unggulan P3SWOT (Peneliti, Penulis, Pencipta, Seniman, Wartawan, Olahragawan, dan Tokoh). Beasiswa P3SWOT ditawarkan bagi peneliti, penulis, pencipta, seniman, wartawan, olahragawan, dan tokoh. Beasiswa ini diberikan sebagai stimulasi bantuan beasiswa untuk lingkup nasional dan internasional dalam rangka menyiapkan para peneliti, pencipta, penulis, seniman, wartawan, olahragawan, dan tokoh yang cerdas dan kompetitif sesuai dengan visi pendidikan nasional (Susanto, dkk., 2010).

Sekretariat Beasiswa Unggulan BPKLN (Biro Perencanaan dan Kerjasama Luar Negeri) Kemdiknas, sebagai penyelenggara program beasiswa unggulan P3SWOT, telah menerapkan pemanfaatan teknologi informasi dalam bentuk *website*, yaitu <http://beasiswaunggulan.kemdiknas.go.id>. *Website* ini menyediakan informasi pendaftaran, formulir pendaftaran yang dapat di-*download* oleh calon pendaftar, dan pengumuman pemenang beasiswa. Namun, pemanfaatan teknologi ini belum optimal, karena *website* yang dikembangkan cenderung bersifat statis. Interaksi yang terjadi dalam *website* tersebut hanya bersifat satu arah, yaitu dari penyedia *website* kepada pengguna.

Mekanisme seleksi beasiswa unggulan P3SWOT yang berjalan saat ini juga tidak terlepas dari sejumlah persoalan, yaitu proses pendaftaran yang kurang efektif dan tidak terintegrasinya data pendaftaran beasiswa unggulan P3SWOT. Pendaftar harus mengirimkan berkas pendaftaran melalui jasa pos yang membutuhkan banyak waktu. Di samping itu, panitia seleksi beasiswa harus mendata berkas pendaftaran yang masuk sebelum melakukan proses seleksi dan penilaian. Untuk mengatasi berbagai persoalan yang masih dihadapi saat ini dan mengantisipasi jumlah pendaftar yang terus meningkat, perlu

dikembangkan sebuah sistem informasi yang dapat menangani manajemen seleksi beasiswa unggulan P3SWOT dengan efektif dan dapat diakses secara *online* sehingga sistem tersebut dapat digunakan kapan dan di mana saja tanpa terbatas pada waktu dan tempat tertentu.

Paradigma pengembangan perangkat lunak saat ini telah berkembang ke konsep *object oriented*. Perangkat lunak berorientasi objek menekankan konsep *reusable* sehingga proses pengembangan perangkat lunak dapat dilakukan lebih cepat dan berkualitas tinggi. Di samping itu, struktur perangkat lunak dapat diuraikan secara inheren serta mudah untuk diadaptasi dan diskalakan, baik untuk *project* skala kecil maupun besar (Pressman, 2001).

Unified Process merupakan salah satu *software development process* yang menerapkan konsep berorientasi objek yang dikembangkan oleh Ivar Jacobson, Grady Booch, dan James Rumbaugh. *Unified Process* merupakan *generic process framework* yang dapat dispesialisasi untuk area aplikasi, tipe organisasi, tingkat kompetensi, dan ukuran proyek yang berbeda (Jacobson, *et al.*, 1999). *Unified Process* bersifat *open, free*, dan tidak terikat dengan *vendor* tertentu (Arlow, *et al.*, 2002). Oleh karena itu, proses pengembangan sistem informasi ini menggunakan *Unified Process* sehingga diharapkan dihasilkan perangkat lunak yang berkualitas tinggi, *reusable*, dan mudah untuk dipelihara.

I.2. Rumusan Masalah

Berdasarkan uraian latar belakang, rumusan permasalahan yang diangkat dalam tugas akhir ini adalah bagaimana mengembangkan Sistem Informasi Seleksi Beasiswa Unggulan P3SWOT Kemdiknas *Online* dengan menggunakan *Unified Process*.

I.3. Tujuan dan Manfaat

Tujuan penyusunan tugas akhir ini adalah menghasilkan sistem informasi yang dapat menangani seleksi beasiswa unggulan P3SWOT Kemdiknas secara *online* menggunakan *Unified Process*.

Adapun manfaat yang didapatkan dari penyusunan tugas akhir ini adalah sebagai berikut :

- 1) Sekretariat Beasiswa Unggulan BPKLN Kemdiknas dapat mengelola seleksi beasiswa unggulan P3SWOT dengan lebih efektif.

- 2) Proses pendaftaran beasiswa unggulan P3SWOT dapat dilakukan dengan lebih mudah menggunakan aplikasi pendaftaran yang dapat diakses secara *online* melalui jaringan Internet.

I.4. Ruang Lingkup

Ruang lingkup tugas akhir ini adalah kegiatan pengembangan Sistem Informasi Seleksi Beasiswa Unggulan P3SWOT Kemdiknas *Online* menggunakan *Unified Process*. Proses pengembangan sistem informasi ini dilakukan melalui fase-fase dalam *Unified Process*, namun penyajian laporan ini distrukturkan berdasarkan *workflow*. Proses seleksi yang ditangani dalam sistem informasi ini mencakup proses pendaftaran beasiswa hingga pelaporan untuk pemenang beasiswa. Sistem informasi ini tidak melakukan pendeteksian keabsahan dokumen yang di-*upload* oleh pendaftar dan tidak menangani administrasi pembiayaan untuk para pemenang beasiswa.

I.5. Sistematika Penulisan

Sistematika penulisan yang digunakan dalam tugas akhir ini terbagi dalam beberapa pokok bahasan, yaitu :

BAB I PENDAHULUAN

Bab ini berisi latar belakang, perumusan masalah, tujuan dan manfaat, ruang lingkup, dan sistematika penulisan dalam pembuatan tugas akhir.

BAB II DASAR TEORI

Bab ini berisi kumpulan studi pustaka yang berhubungan dengan topik tugas akhir. Dasar teori yang digunakan dalam penyusunan tugas akhir ini meliputi sistem informasi, beasiswa unggulan P3SWOT Kemdiknas, konsep *object oriented*, *Unified Modeling Language (UML)*, *Unified Process*, bahasa pemrograman PHP, *framework* CodeIgniter, dan sistem manajemen basis data MySQL.

BAB III DEFINISI KEBUTUHAN, ANALISIS, DAN PERANCANGAN

Bab ini membahas proses pengembangan perangkat lunak dan hasil yang didapatkan pada tahap definisi kebutuhan, analisis, dan perancangan.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas proses pengembangan perangkat lunak dan hasil yang didapat pada tahap implementasi. Bab ini juga berisi rincian pengujian perangkat lunak yang dibangun dengan metode *black box*.

BAB V PENUTUP

Bab ini berisi kesimpulan yang diambil berkaitan dengan perangkat lunak yang dikembangkan dan saran-saran untuk pengembangan perangkat lunak lebih lanjut.

BAB II

DASAR TEORI

Pada bab ini disajikan dasar teori yang digunakan dalam penyusunan tugas akhir ini, yaitu sistem informasi, beasiswa unggulan P3SWOT Kemdiknas, konsep *object oriented*, *Unified Modeling Language* (UML), *Unified Process*, bahasa pemrograman PHP, *framework* CodeIgniter, dan sistem manajemen basis data MySQL.

II.1. Sistem Informasi

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Setiap sistem mempunyai tiga elemen utama, yaitu *input*, transformasi, dan *output*. Data terdiri atas fakta-fakta dan angka-angka yang relatif tidak berarti bagi pengguna, sedangkan informasi adalah data yang telah diproses dan memiliki arti bagi penggunanya. Dengan demikian, sistem informasi merupakan sekumpulan elemen yang terintegrasi yang digunakan untuk mengubah data menjadi informasi (McLeod, 2001).

II.2. Beasiswa Unggulan P3SWOT Kemdiknas

Beasiswa unggulan P3SWOT (Peneliti, Penulis, Pencipta, Seniman, Wartawan, Olahragawan, dan Tokoh) merupakan salah satu beasiswa unggulan yang diberikan oleh Sekretariat Beasiswa Unggulan BPKLN (Biro Perencanaan dan Kerjasama Luar Negeri) Kementerian Pendidikan Nasional (Kemdiknas) bagi peneliti, penulis, pencipta, seniman, wartawan, olahragawan, dan tokoh. Beasiswa ini diberikan sebagai stimulasi bantuan beasiswa untuk lingkup nasional dan internasional dalam rangka menyiapkan para peneliti, pencipta, penulis, seniman, wartawan, olahragawan, dan tokoh yang cerdas dan kompetitif sesuai dengan visi pendidikan nasional. Program beasiswa unggulan P3SWOT ini diharapkan dapat melahirkan *critical mass* para peneliti, penulis, pencipta, seniman, wartawan, olahragawan, dan tokoh yang berdaya saing tinggi pada masa yang akan datang (Susanto, dkk., 2010).

Pendaftar beasiswa unggulan P3SWOT yang lolos seleksi berhak mendapatkan sejumlah dana sesuai dengan anggaran yang telah ditetapkan. Dana tersebut dapat digunakan oleh para penerima beasiswa untuk membiayai kegiatan yang telah diusulkan

dan disepakati dalam kontrak. Setelah selesai melakukan kegiatan tersebut, penerima beasiswa wajib menyerahkan laporan kegiatan yang berisi uraian mengenai pelaksanaan kegiatan kepada Sekretariat Beasiswa Unggulan BPKLN Kemdiknas.

Prosedur seleksi beasiswa unggulan P3SWOT yang berjalan saat ini adalah sebagai berikut :

- 1) Pendaftar mengirimkan sejumlah dokumen atau persyaratan untuk mendaftar beasiswa unggulan P3SWOT ke Sekretariat Beasiswa Unggulan BPKLN Kemdiknas. Dokumen atau persyaratan tersebut terdiri atas:
 - a) Formulir pendaftaran beasiswa unggulan P3SWOT (*draft* dapat diunduh di beasiswaunggulan.kemdiknas.go.id).
 - b) Proposal kegiatan yang berisi uraian tentang kegiatan yang akan dilakukan sesuai kriteria profesi yang dipilih dan anggaran dana yang dibutuhkan.
 - c) Fotokopi buku tabungan halaman pertama yang terdapat nomor rekening, diutamakan atas nama lembaga tempat pendaftar bernaung.
 - d) Kontrak kerjasama sebagai bukti persetujuan permohonan yang diusulkan (*draft* dapat diunduh beasiswaunggulan.kemdiknas.go.id).
 - e) Kuitansi penerimaan dana bila permohonan disetujui (*draft* dapat diunduh di beasiswaunggulan.kemdiknas.go.id).
 - f) Dokumen-dokumen lain untuk melengkapi usulan beasiswa, antara lain:
 - i) Surat undangan kegiatan ilmiah, seminar, *workshop*, *short course*, atau kegiatan profesionalisme ke luar negeri.
 - ii) *Curriculum Vitae* (CV) atau biodata.
 - iii) Abstrak dari kegiatan seminar, *workshop*, atau *short course* yang diusulkan untuk didanai (untuk pendaftar beasiswa dalam kategori peneliti, penulis, pencipta, seniman, atau olahragawan)
 - iv) Karya, ciptaan, buku, atau tulisan yang telah dihasilkan selama 3 tahun terakhir.
 - v) Tulisan dalam bentuk buku yang siap diterbitkan ke penerbit nasional atau internasional (untuk pendaftar beasiswa dalam kategori penulis, pencipta, atau peneliti).
 - vi) Tulisan, koran, atau majalah yang mencantumkan ketokohan (untuk pendaftar beasiswa dalam kategori tokoh)

- 2) Petugas seleksi beasiswa unggulan P3SWOT melakukan penilaian terhadap berkas pendaftaran yang masuk, kemudian menetapkan pendaftar yang berhak menjadi penerima (pemenang) beasiswa unggulan P3SWOT. Penetapan pemenang dilakukan berdasarkan sejumlah kriteria, yaitu kelengkapan syarat administrasi, prestasi pendaftar, proposal kegiatan yang diajukan, dan rekomendasi yang diberikan kepada pendaftar.
- 3) Pengumuman pemenang beasiswa unggulan P3SWOT dapat dilihat di *website* beasiswa unggulan. Pemenang beasiswa yang telah ditetapkan tersebut akan mendapatkan SK (Surat Keputusan) dan sejumlah dana dari Sekretariat Beasiswa Unggulan BPKLN Kemdiknas untuk membiayai pelaksanaan kegiatan yang telah disepakati dalam kontrak.
- 4) Setelah kegiatan selesai dilaksanakan, penerima beasiswa unggulan P3SWOT wajib mengirimkan laporan kegiatan ke Sekretariat Beasiswa Unggulan BPKLN Kemdiknas sesuai kontrak kerjasama yang ditandatangani.

II.3. Konsep *Object Oriented*

Konsep *object oriented* memfokuskan pada penciptaan *class* yang merupakan *blueprint* dari suatu objek. Konsep ini membagi perangkat lunak menjadi beberapa objek yang saling berinteraksi antara satu dengan lainnya. Beberapa istilah yang berkaitan dengan konsep *object oriented* adalah :

1) *Class* dan Objek

Class dapat diartikan deskripsi secara umum (*template*, *pattern*, atau *blueprint*) yang menggambarkan sekumpulan objek yang serupa (Pressman, 2001). Objek dapat berupa objek fisik seperti meja atau pelanggan maupun objek konseptual seperti *text input area* atau *file* (Welling, *et al.*, 2004).

2) Atribut, *Method* dan *Message*

Atribut adalah sesuatu yang melekat pada objek yang mendeskripsikan sifat *class* atau objek. Sebuah objek mengenkapsulasi data (direpresentasikan sebagai kumpulan atribut) dan algoritma yang memproses data tersebut. Algoritma ini disebut operasi, *method*, atau *service*. Setiap operasi yang dienkapsulasi oleh sebuah objek memberikan representasi salah satu *behavior* dari objek tersebut.

Suatu objek berinteraksi dengan objek lainnya melalui *message*. Sebuah objek diminta untuk melakukan salah satu operasinya dengan mengirimkannya sebuah *message*.

Objek penerima merespon *message* tersebut dengan memilih operasi yang mengimplementasikan nama *message*, mengeksekusi operasi, dan kemudian mengembalikan kontrol kepada objek yang memanggil (Pressman, 2001).

3) Enkapsulasi

Sebuah *class* mengenkapsulasi data dan operasi yang memproses data tersebut. Data (atribut) yang menggambarkan kelas ditutup oleh operasi yang memanipulasi data tersebut. Untuk mengakses nilai atribut *class* harus melalui sebuah operasi. Konsep enkapsulasi ini mendukung *information hiding*. Detail implementasi internal dari data dan prosedur disembunyikan dari dunia luar. Hal ini mengurangi efek samping ketika terjadi perubahan dalam *class* (Pressman, 2001).

4) *Inheritance*

Inheritance merupakan pewarisan sifat dari sebuah *class* ke *class* yang baru. *Subclass* Y merupakan pewaris dari *superclass* X, maka *subclass* Y mewarisi semua atribut dan operasi yang dimiliki oleh *superclass* X. Hal ini mendukung konsep *reuse*. Pada setiap level hirarki *class*, atribut dan operasi baru dapat ditambahkan ke *class* yang telah diwarisi dari level yang lebih tinggi dalam hirarki. Pada *inheritance* juga memungkinkan terjadinya *overriding*. *Overriding* terjadi ketika atribut dan operasi yang diwarisi, dimodifikasi untuk kebutuhan spesifik dari *class* yang baru (Pressman, 2001).

5) Polimorfisme

Polimorfisme mengizinkan sejumlah operasi yang berbeda untuk mempunyai nama yang sama. Hal ini membuat objek saling terpisah dari objek lainnya dan membuat setiap objek lebih independen (Pressman, 2001).

II.4. *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa standar yang digunakan untuk menulis *blueprint* perangkat lunak. UML dapat digunakan untuk memvisualisasi, menspesifikasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. UML hanya merupakan sebuah bahasa, sehingga UML hanya salah satu bagian dari metode pengembangan perangkat lunak (Booch, *et al.*, 2005). UML terdiri atas 3 *building block*, yaitu *things*, *relationship*, dan diagram.

1) *Things*

Things adalah *building block* berbasis objek yang utama dari UML. *Things* terdiri atas 4 macam, yaitu :

a) *Structural Things*

Structural things adalah bagian model statis yang merepresentasikan elemen konseptual atau fisik. Jenis-jenis *structural things* adalah *class*, *interface*, *collaboration*, *use case*, *active class*, *component*, *artifact*, dan *node* (Booch, et al., 2005).

b) *Behaviorial Things*

Behaviorial things adalah bagian dinamis dari model UML yang merepresentasikan *behavior*. Jenis *behaviorial things* adalah *interaction*, *state machine*, dan *activity* (Booch, et al., 2005).

c) *Grouping Things*

Grouping things adalah bagian organisasional dari model UML. Satu-satunya jenis *grouping things* adalah *package*. *Package* merupakan mekanisme untuk mengorganisasikan *design* itu sendiri. *Structural things*, *behaviorial things*, dan bahkan *grouping things* yang lain dapat ditempatkan dalam sebuah *package*. (Booch, et al., 2005).

d) *Annotational Things*

Annotational things adalah bagian penjelas dari model UML. Jenis *annotational things* hanya satu, yaitu *note* yang digunakan untuk memberikan *constraint* atau komentar yang dilekatkan pada sebuah atau sekumpulan elemen.

2) *Relationship*

Relationship merupakan *building block* UML yang berfungsi sebagai penghubung antar-*things*. Jenis *relationship* yang terdapat pada UML adalah *dependency*, *association*, *generalization*, dan *realization* (Booch, et al., 2005).

3) *Diagram*

Diagram adalah presentasi grafis dari sekumpulan elemen. Diagram merupakan kombinasi antara *things* dan *relationship*. Diagram dibuat untuk memvisualisasikan sistem dari sejumlah perspektif yang berbeda, sehingga diagram merupakan proyeksi terhadap sistem. Pada UML versi 2 terdapat 13 diagram, yaitu: *class*, *object*,






component, composite structure, use case, sequence, communication, state, activity, deployment, package, timing, dan interaction diagram (Booch, et al., 2005).

Penjelasan sejumlah diagram yang umum digunakan adalah sebagai berikut:

a) *Class diagram*

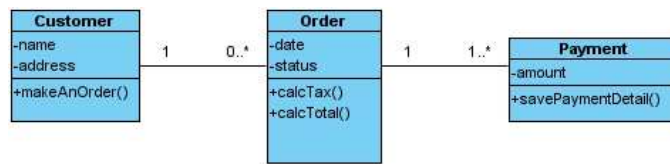
Class diagram menggambarkan sejumlah *class* dan hubungan antar-*class* tersebut di dalam sistem. Selama *design*, *class diagram* digunakan untuk meng-*capture* struktur *class* yang membangun arsitektur sistem. Dua elemen utama dari *class diagram* adalah *class* dan *relationship*. Notasi *class* terdiri atas 3 bagian, yaitu nama, atribut, dan operasi. Nama suatu *class* digunakan sebagai pengenalan. Atribut merupakan informasi yang dimiliki oleh suatu *class*, sedangkan operasi merupakan tingkah laku yang didefinisikan oleh suatu *class* (Booch, et al., 2007). *Class* dapat berkolaborasi dengan *class* lainnya dalam berbagai cara. Jenis *relationship* antar-*class* dapat dilihat pada Tabel 2.1.

Tabel 2.1 Jenis *Relationship* pada *Class Diagram*

Jenis	Deskripsi	Gambar
<i>Association</i>	<i>Association</i> menghubungkan dua <i>class</i> dan menunjukkan hubungan semantik. <i>Association</i> dapat mempunyai arah yang menunjukkan <i>navigability</i> .	
<i>Dependency</i>	<i>Dependency</i> adalah relasi yang menunjukkan bahwa sebuah <i>class</i> menggunakan informasi atau layanan dari <i>class</i> lain.	
<i>Generalization</i>	<i>Generalization</i> menunjukkan hubungan generalisasi/spesialisasi.	
<i>Aggregation</i>	<i>Aggregation</i> menunjukkan hubungan "bagian dari" atau <i>whole/part hierarchy</i> .	
<i>Composition</i>	<i>Composition</i> menunjukkan jenis khusus dari <i>aggregation</i> dengan bagian bawah <i>multiplicity</i> selalu bernilai 1.	

Dalam *class diagram* dikenal adanya *multiplicity*, yaitu angka yang menunjukkan jumlah kemungkinan instantiasi dari sebuah *class* berhubungan dengan instantiasi dari *class* yang lain. *Multiplicity* dapat dilihat dari angka yang terdapat pada hubungan antar-*class*, misalnya 1, 0..1, 0..*, dan 1..*. Pada umumnya *multiplicity* didefinisikan dengan batas bawah dan batas atas. Batas bawah dapat berupa angka positif atau nol, sedang batas atas dapat berupa angka positif atau * (untuk nilai tak

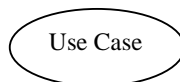
terhingga). Jika batas bawah dan batas atas bernilai sama maka hanya perlu menuliskan sekali. Contoh *class diagram* dapat dilihat pada Gambar 2.1.



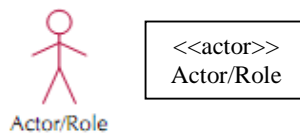
Gambar 2.1 Contoh *Class Diagram*

b) *Use case diagram*

Use case diagram menunjukkan sekumpulan *use case*, aktor, dan *relationship* yang terjadi antara *use case* dan aktor tersebut. *Use case* menggambarkan bagaimana aktor berinteraksi dengan sistem. Aktor adalah entitas eksternal yang memainkan peranan (*role*) ketika berinteraksi dengan sistem. Aktor dapat berupa *user* atau benda seperti *dumb terminal*, sensor, atau sistem komputer lainnya (Hunt, 2003). Simbol *use case* dan aktor dapat dilihat pada Gambar 2.2 dan 2.3. Aktor manusia disimbolkan dengan *stick figure*, sedangkan *non-human actor* (benda) disimbolkan menggunakan segi empat dengan *stereotype* <<actor>> (Dennis, *et al.*, 2005).



Gambar 2.2 Simbol *Use Case*



Gambar 2.3 Simbol Aktor

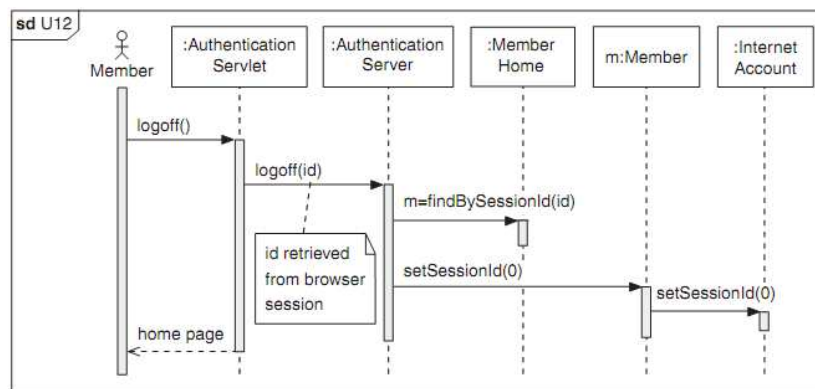
Jenis *relationship* pada *use case* dapat dilihat pada Tabel 2.2 (Booch, *et al.*, 2005).

Tabel 2.2 Jenis *Relationship* pada *Use Case*

Jenis	Deskripsi	Gambar
<i>Association</i>	<i>Association</i> merupakan komunikasi antara aktor dan <i>use case</i> yang terlibat.	_____
<i>Generalization</i>	<i>Generalization</i> merupakan hubungan generalisasi/spesialisasi. <i>Generalization</i> dapat terjadi antar <i>use case</i> atau aktor.	_____▷
<i>Extend</i>	Relasi yang menspesifikasikan bahwa <i>target use case</i> meng- <i>extend behavior</i> dari <i>source use case</i> .	- - - - -▷ <<extends>>
<i>Include</i>	X <i>include</i> Y berarti <i>use case</i> X menggunakan <i>use case</i> Y sepenuhnya.	◀ - - - - - <<include>>

c) *Sequence diagram*

Sequence diagram adalah sebuah *interaction diagram* yang menekankan urutan waktu pertukaran pesan. *Sequence diagram* terdiri atas 2 bagian utama, yaitu *lifeline* dan *message*. *Lifeline* digambarkan dengan *dashed vertical line* yang digambar di bawah objek. *Lifeline* menunjukkan masa hidup (*existence*) objek. *Message* digambarkan dengan anak panah antara 2 garis vertikal (*lifeline* objek). Urutan *message* ditunjukkan secara vertikal. *Message* pertama digambarkan paling atas, sedangkan *message* terakhir digambarkan paling bawah dalam diagram (Booch, *et al.*, 2007).



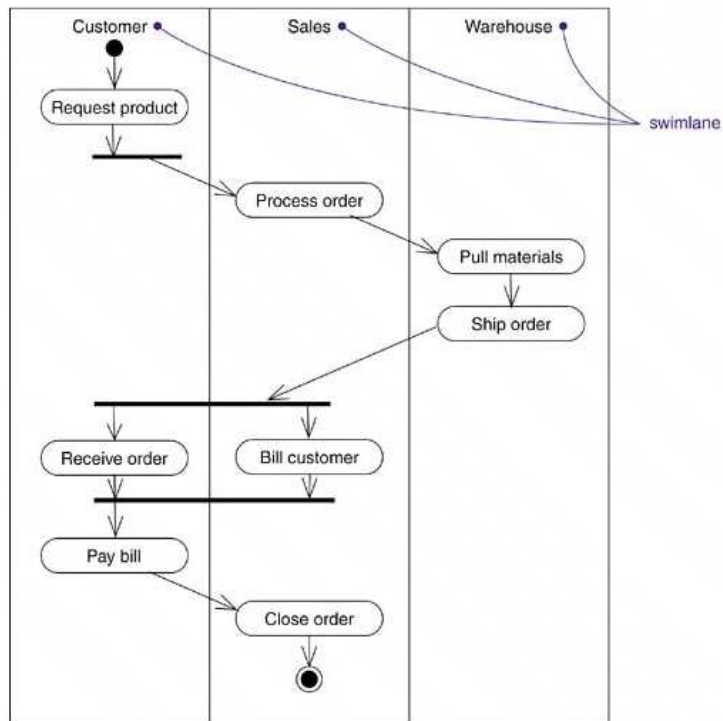
Gambar 2.4 Contoh *Sequence Diagram*

Sequence diagram pada Gambar 2.4 menjelaskan bagaimana Member dapat log off dari sistem. Member meminta `AuthenticationServlet` untuk `logoff`. `AuthenticationServlet` meneruskan *request* ke `AuthenticationServer`. `AuthenticationServer` membaca id dari *browser session*. Selanjutnya, `AuthenticationServer` menemukan hubungan dengan objek Member dan meminta objek Member untuk mengeset session id menjadi 0. Objek Member meneruskan request ke `InternetAccount`. Akhirnya home page disajikan kepada Member (O'Docherty, 2005).

d) *Activity diagram*

Activity diagram menggambarkan berbagai aliran aktivitas yang terjadi di dalam sistem, titik awal dari masing-masing aliran, keputusan yang mungkin terjadi, dan akhir dari aliran aktivitas tersebut. Diagram ini dilengkapi dengan alur percabangan, kondisional, serta sinkronisasi (untuk aktifitas yang dilakukan secara konkuren) untuk menjelaskan aliran aktivitas di dalam sistem. *Activity diagram*

dapat digunakan untuk memodelkan *workflow* proses bisnis. Untuk membagi aktivitas bisnis ke dalam kelompok-kelompok tertentu sesuai dengan tanggung jawabnya dalam organisasi dapat digunakan notasi *swimlane*. *Swimlane* dapat merepresentasikan entitas di dunia nyata seperti unit organisasional dalam sebuah perusahaan. Dalam sebuah *activity diagram* yang dipartisi ke dalam beberapa *swimlane*, setiap aktivitas hanya dapat berada pada satu *swimlane*, tetapi transaksi dapat terjadi antar *lane*. Contoh *activity diagram* yang menggunakan *swimlane* dapat dilihat pada gambar 2.5 (Booch, *et al.*, 2005).



Gambar 2.5 Contoh Activity Diagram Menggunakan Swimlane

II.5. Unified Process

Unified Process merupakan salah satu *software development process* yang telah mendukung konsep berorientasi objek. *Unified Process* mempunyai tiga karakteristik utama, yaitu (Jacobson, *et al.*, 1999) :

1) Use Case Driven

Use case tidak hanya merupakan *tool* untuk menspesifikasi kebutuhan fungsional sistem, tetapi *use case* juga mengarahkan semua *workflow* dalam proses pengembangan *software*. Model *design* dan implemetasi yang merealisasikan *use case* serta sejumlah *test case* untuk pengujian dibuat berdasarkan *use case model*. (Jacobson, *et al.*, 1999).

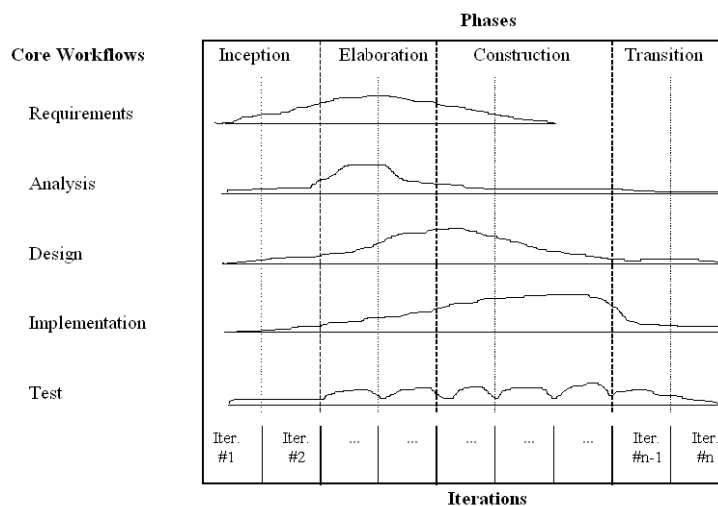
2) *Architecture Centric*

Arsitektur merupakan pandangan terhadap keseluruhan *design* yang menonjolkan karakteristik-karakteristik yang penting dan meninggalkan sisi *detail*-nya. Arsitektur dan *use case* haruslah seimbang. Realisasi *use case* harus disesuaikan dengan arsitektur yang dikembangkan, sebaliknya arsitektur juga harus memberikan ruang untuk realisasi semua *use case* yang dibutuhkan, baik pada awal pengembangan hingga beberapa waktu ke depan (Jacobson, *et al.*, 1999).

3) *Iteration & Incremental*

Iterasi mengacu pada tahapan-tahapan dalam *workflow*, sedangkan *increment* mengacu pada pertumbuhan produk. Proses pengembangan *software* melalui iterasi sekumpulan *workflow* yang diorganisasikan ke dalam empat fase, yaitu *inception*, *elaboration*, *construction*, dan *transition*. Setiap iterasi akan memberikan pengaruh pada kemajuan sistem (Jacobson, *et al.*, 1999).

Dalam *Unified Process*, *project* dikembangkan dalam empat fase yang berbeda. Saat *project* beralih dari satu fase ke fase berikutnya, jumlah kerja yang dilakukan di tiap *workflow* juga berubah. Gambar 2.6 menunjukkan hubungan antara fase dengan *workflow*. Bagian horizontal atas menunjukkan fase-fase yang harus dijalani, bagian vertikal ke bawah menunjukkan *workflow*, sedangkan bagian horizontal bawah menunjukkan iterasi. Kurva menunjukkan bobot kerja yang harus dilakukan pada setiap *workflow* dalam setiap fase. Setiap fase dapat terdiri atas satu atau beberapa iterasi *workflow*. Banyaknya iterasi pada setiap fase tergantung pada kompleksitas sistem yang dikembangkan (Jacobson, *et al.*, 1999).



Gambar 2.6 Hubungan Fase dengan *Workflow* dalam *Unified Process*

Penjelasan tiap-tiap fase dalam pengembangan *software* adalah sebagai berikut (Jacobson, *et al.*, 1999):

1) *Inception*

Inception mendefinisikan lingkup *project* dan mengembangkan *business case* untuk sistem. Fase ini terfokus pada *requirement* serta sedikit melakukan analisis dan *design*.

2) *Elaboration*

Aktivitas yang dilakukan pada fase ini adalah melengkapi *requirement*, baik fungsional maupun nonfungsional serta melakukan analisis dan *design*. Salah satu aktivitas utama dalam fase ini adalah pembuatan arsitektur sistem. Pada fase ini, juga telah dilakukan sedikit implementasi dan pengujian untuk mengeksekusi arsitektur yang diciptakan tersebut.

3) *Construction*

Construction merupakan fase pembangunan sistem. Pada fase ini masih dapat dilakukan sedikit analisis untuk melengkapi hasil analisis yang telah dihasilkan pada fase sebelumnya. Tiga *workflow* terakhir, yaitu *design*, implementasi, dan pengujian menjadi aktivitas utama dalam fase ini.

4) *Transition*

Transition merupakan peralihan produk ke lingkungan *user* (*beta release*). Sejumlah *user* akan mencoba produk yang telah dihasilkan dan melaporkan *defect* yang ditemukan. Fase ini berakhir pada *release product*.




Setiap fase dapat terdiri atas satu atau beberapa iterasi *workflow* (*requirement*, *analysis*, *design*, *implementation*, dan *test*). *Workflow* menunjukkan urutan aktivitas yang harus dilakukan, dan aktivitas menentukan apa yang harus dilakukan dan artefak yang perlu dihasilkan (Jacobson, *et al.*, 1999). Aktivitas yang dilakukan dalam tiap *workflow* tersebut adalah sebagai berikut :

1) *Requirement*

Tujuan dari *requirement* adalah untuk menemukan dan mencapai persetujuan mengenai apa yang harus dilakukan sistem yang diungkapkan dalam bahasa *user* (Arlow, *et al.*, 2002). Sebelum meng-*capture requirement*, dapat dilakukan pemodelan bisnis terlebih dahulu untuk mengetahui konteks sistem. Pemodelan bisnis adalah teknik untuk memahami proses bisnis dalam sebuah organisasi. Pemodelan bisnis didukung oleh model UML dengan menggunakan *business use case model*. *Business use case model*

menggambarkan proses bisnis organisasi dalam istilah *business use case* serta *business worker* dan *business actor* yang berkorespondensi untuk *business use case* tersebut (Jacobson, *et al.*, 1999). Notasi yang digunakan untuk dalam *business use case model* disajikan pada Tabel 2.3 (Kruchten, 2003).

Tabel 2.3 Notasi *Business Use Case Model*

Jenis	Deskripsi	Gambar
<i>Business actor</i>	Aktor di luar organisasi yang berinteraksi dengan organisasi dan terlibat dalam kegiatan bisnis organisasi.	
<i>Business use case</i>	Model yang digunakan untuk menggambarkan proses bisnis organisasi.	
<i>Business worker</i>	Suatu peranan di dalam organisasi.	

Requirement sistem dapat diidentifikasi dari *business use case model* yang telah dibuat. *Output* dari *requirement* ini berupa *use case model*. *Use case model* terdiri atas aktor, *use case*, dan artifak lain seperti *GUI prototype* dan *non-functional requirement* (Hunt, 2003). Aktor sistem dapat diidentifikasi dari *business worker* dan *business actor* yang akan berinteraksi dengan sistem. *Use case* dapat diperoleh dengan mengidentifikasi bagaimana sistem mendukung *business worker* dan *business actor* dalam menjalankan proses bisnis. Selanjutnya, *use case* disempurnakan untuk memenuhi kebutuhan aktor selengkapnya (Jacobson, *et al.*, 1999).

2) Analisis

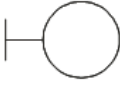


Analisis dilakukan untuk menerjemahkan *requirement* ke dalam bahasa *developer* dan mengidentifikasi elemen/entitas utama dari sistem yang diperlukan untuk memenuhi *user requirement*. Pada *workflow* ini *requirement* yang telah diidentifikasi dalam *use case* dianalisis dan disusun dalam lingkup internal sistem (Hunt, 2003).

Artifak-artifak yang dihasilkan dalam analisis antara lain (Jacobson, *et al.*, 1999) :

a) *Analysis class*

Analysis class digunakan untuk merepresentasikan satu atau beberapa *class* dan/atau subsistem dalam lingkup *design* sistem. *Analysis class* digambarkan dengan menggunakan tiga *stereotype* dasar, yaitu *entity*, *boundary*, dan *control*. Penjelasan dari masing-masing *analysis class* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Jenis *Analysis Class*

Jenis	Deskripsi	Gambar
<i>Boundary</i>	<i>Boundary class</i> digunakan untuk menggambarkan interaksi antara sistem dengan aktor. Setiap <i>boundary class</i> harus dihubungkan minimal dengan satu aktor.	
<i>Control</i>	<i>Control class</i> digunakan untuk mengkoordinasikan interaksi antara aktor (melalui <i>boundary class</i>) dengan data di <i>entity class</i> .	
<i>Entity</i>	<i>Entity class</i> digunakan untuk merepresentasikan <i>long-lived persistent data</i> .	

b) *Use case realization – analysis*

Use case realization – analysis digunakan untuk menghubungkan *use case* dengan *analysis class* dengan menggambarkan *class* mana yang berpartisipasi dalam implementasi *use case* dan interaksi yang terjadi antar-*class* tersebut (Hunt, 2003).

c) *Analysis package*

Analysis package berfungsi untuk mengorganisasikan artifak dalam model analisis ke dalam bagian-bagian yang lebih teratur. *Analysis package* dapat berisi *analysis class*, *use case realization*, dan *analysis package* lainnya (rekursif). *Analysis package* seharusnya bersifat kohesif, yaitu isi di dalamnya harus berkaitan erat dan bersifat *loosely-coupled*, yaitu ketergantungan dengan *package* lainnya rendah (Jacobson, *et al.*, 1999).

3) *Design*

Design dilakukan untuk menspesifikasikan secara penuh bagaimana fungsionalitas diimplementasikan dengan menggunakan model yang telah dihasilkan dari *workflow* analisis. Artifak yang dihasilkan dalam *workflow* ini meliputi (Arlow, *et al.*, 2002) :

a) *Design subsystem*

Design subsystem digunakan untuk mengorganisasikan artifak dalam *design model* ke dalam bagian-bagian yang lebih teratur. Artifak ini terdiri atas *design classes*, *use case realization*, *interface*, dan subsistem lainnya.

b) *Design classes*

Design classes digunakan untuk menggambarkan *class* yang akan diimplementasi beserta spesifikasi atribut dan operasi, agregasi dan asosiasi, serta generalisasi.

c) *Use case realization - design*

Use case realization - design mendeskripsikan bagaimana suatu *use case* di-*design* dalam kolaborasi dengan *design model*. *Use case realization* terdiri atas *design interaction diagram* yang dapat dimodelkan dengan *sequence diagram* untuk tiap *use case* dan *class diagram* yang berisi *design class* yang terlibat pada suatu *use case*.

Pada alur kerja inti *design* juga dibuat *database design* yang digunakan sebagai tempat penyimpanan data. *Database* yang digunakan dalam pengembangan aplikasi ini adalah *relational database*. *Relational database design* merupakan *mapping* ke dalam *relational table* untuk setiap *class* dalam *class diagram* yang bersifat *persistent*. Berikut ini adalah dasar dalam memetakan objek ke dalam *relational database* (Ambler, 2000):

a) Memetakan atribut ke dalam kolom

Sebuah atribut dalam objek akan dipetakan menjadi beberapa kolom atau tidak sama sekali.

b) Memetakan *class* ke dalam tabel

Setiap *class* dalam *class diagram* menjadi sebuah tabel. Sedangkan untuk mengimplementasikan *inheritance* ke dalam *relational database*, terdapat tiga strategi yaitu : menggunakan satu tabel untuk setiap *class hierarchy*, menggunakan satu tabel untuk setiap *concrete class*, dan menggunakan satu tabel untuk setiap *class*. Dalam pengembangan aplikasi ini strategi yang digunakan adalah strategi yang kedua, yaitu menggunakan satu tabel untuk setiap *concrete class*. Dengan strategi ini, setiap tabel memperoleh warisan atribut dari *parent class*.

c) Memetakan *relationship*

Relationship juga harus dipetakan ke dalam *database*. *Relationship* dalam *relational database* berhubungan dengan penggunaan *foreign key*. *Foreign key* ini merupakan atribut dalam suatu tabel yang mengacu ke tabel lain.

4) *Implementation*

Implementation dilakukan untuk mentransformasikan *design model* ke dalam *executable code*. Bahasa pemrograman yang digunakan untuk implementasi perangkat lunak ini adalah PHP 5. PHP 5 telah menyediakan implementasi konsep *object oriented*

yang *powerful* dan mempunyai fitur-fitur yang lengkap untuk bahasa pemrograman yang berorientasi objek (Welling, *et al.*, 2004). DBMS yang digunakan adalah MySQL 5. Pada *workflow implementation* ini dilakukan implementasi *component*, subsistem, dan *interface* (Arlow, *et al.*, 2002).

5) *Test* (Pengujian)

Tujuan dari pengujian adalah untuk memastikan bahwa sistem dapat menyediakan fungsionalitas yang diperlukan. Pengujian yang dilakukan hanya pengujian secara *black box*, yaitu menguji fungsionalitas dari perangkat lunak, tanpa harus mengetahui struktur internal program (Pressman, 2001). Artifak yang dihasilkan pada *workflow* ini adalah *test model*, yang mendeskripsikan bagaimana *executable component* diuji oleh *integration test* dan *system test*. *Test model* terdiri atas sejumlah *test case*. Setiap *test case* menspesifikasi apa yang akan diuji dalam sistem (Hunt, 2003).

II.6. Bahasa Pemrograman PHP *Hypertext Presprocessor*

PHP adalah bahasa *scripting* yang bersifat *open source* dan dapat disatukan ke kode HTML. Kode PHP dijalankan di *server* dan menghasilkan HTML yang dikirimkan ke *client*. *Client* akan menerima hasil eksekusi *script*, tanpa mengetahui apa yang dijalankan di *server*. PHP sangat tepat digunakan untuk pengembangan aplikasi berbasis web. (Achour, 2008). PHP versi terbaru adalah PHP 5. Bahasa pemrograman PHP mempunyai banyak keunggulan, antara lain (Welling, *et al.*, 2004) :

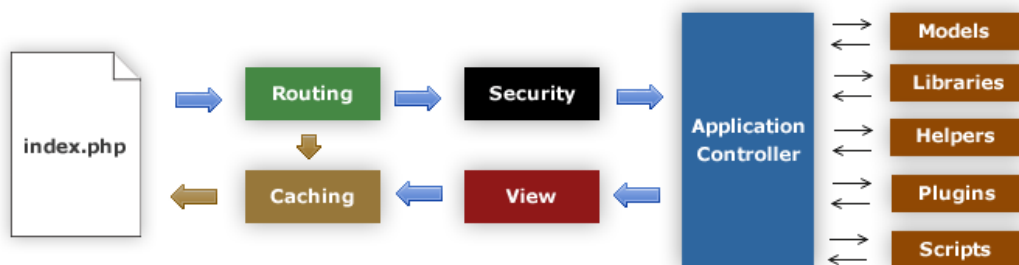
- 1) PHP sangat efisien sehingga dapat memberikan *high performance*.
- 2) PHP adalah bahasa pemrograman yang *free* dan *open source*.
- 3) PHP mempunyai antarmuka ke banyak sistem manajemen basis data seperti MySQL, PostgreSQL, mSQL, Oracle, Hyperware, Sybase, Informix, dan sebagainya.
- 4) PHP mempunyai *built-in libraries* yang menyediakan banyak fungsi untuk menyelesaikan tugas yang berhubungan dengan web.
- 5) PHP dapat dijalankan pada banyak sistem operasi, seperti Linux, FreeBSD, Solaris, IRIX, maupun Microsoft Windows.
- 6) PHP 5 telah menyediakan implementasi konsep *object oriented* yang *powerful* dan mempunyai fitur-fitur yang lengkap untuk bahasa pemrograman yang berorientasi objek .

II.7. Framework CodeIgniter

Codeigniter adalah sebuah *framework* pengembangan aplikasi (*application development framework*), yaitu sebuah *toolkit* untuk pengguna yang akan membangun aplikasi berbasis web menggunakan PHP. CodeIgniter merupakan *open source framework* yang terlisensi dibawah Apache/BSD-style, sehingga dapat digunakan oleh siapa saja secara bebas. CodeIgniter bertujuan untuk membantu *developer* agar dapat mengembangkan proyek lebih cepat dibandingkan jika harus menulis *code* dari awal, dengan menyediakan sekumpulan *library* untuk kebutuhan-kebutuhan umum, yang di-*design* dalam *interface* dan struktur logik yang sederhana untuk mengakses *library* tersebut. Dengan demikian, *developer* akan lebih terfokus pada proyek dengan meminimalkan jumlah kode yang dibutuhkan untuk menyelesaikan suatu tugas (Ellislab, 2009).

CodeIgniter berdasarkan *Model-View-Controllers (MVC) development pattern*. MVC adalah sebuah pendekatan *software* yang mengijinkan pemisahan antara *logic* dan *presentation*. Dalam praktiknya, halaman *web* akan berisi *script* yang lebih sedikit karena *presentation* dipisahkan dari *scripting* PHP.

Model merepresentasikan struktur data. Umumnya kelas model berisi fungsi-fungsi yang membantu *developer* untuk melakukan *retrieve*, *insert*, dan *update* data di dalam basis data. *View* adalah informasi yang dipresentasikan kepada *user*. Sebuah *view* umumnya berupa halaman *web*, tetapi dalam CodeIgniter sebuah *view* juga dapat berupa bagian dari halaman *web*, seperti *header* atau *footer*. *View* juga dapat berupa halaman RSS atau tipe halaman lainnya. *Controller* bertindak sebagai penghubung antara model, *view*, dan *resource* lain yang diperlukan untuk memproses HTTP *request* dan men-*generate* sebuah halaman *web* (Ellislab, 2009). Gambar 2.7 mengilustrasikan *flow chart* dalam aplikasi yang menggunakan *framework* CodeIgniter (Ellislab, 2009).



Gambar 2.7 Flow Chart Aplikasi dalam CodeIgniter

Penjelasan dari *flow chart* aplikasi pada Gambar 2.6 adalah sebagai berikut:

- 1) *File index.php* bertindak sebagai *front controller*, menginisialisasi *resource* dasar yang diperlukan untuk menjalankan CodeIgniter.
- 2) Router memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan.
- 3) Jika sebuah *cache file* sudah ada, halaman *web* akan langsung dikirim kepada *browser*, tidak melalui eksekusi sistem secara normal.
- 4) *Security*. Sebelum *controller* aplikasi di-load, *HTTP request* dan data yang di-submit oleh *user* di-filter terlebih dahulu untuk menjamin keamanan.
- 5) *Controller* me-load *model*, *library* utama, *plugin*, *helper*, dan *resource* lain yang diperlukan untuk memproses *request*.
- 6) Selanjutnya, *controller* me-load *view* dan mengirimkannya kepada *browser* agar dapat dilihat oleh *user*. Jika *caching* diaktifkan, *view* akan di-cache terlebih dahulu, sehingga pada *request* selanjutnya *view* dapat langsung dikirimkan kepada *browser* tanpa harus me-load dari awal.

II.8. Sistem Manajemen Basis Data MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*) dan bersifat *open source*. Sistem manajemen basis data MySQL menggunakan kumpulan perintah sederhana untuk memasukkan, memanggil, menghapus, dan memperbarui data, yang memungkinkan untuk dapat mengembangkan *database* yang kompleks. Beberapa keunggulan MySQL adalah sebagai berikut (Valade, 2004) :

- 1) MySQL memiliki kecepatan yang bagus dan berukuran kecil.
- 2) MySQL dapat berjalan pada berbagai sistem operasi seperti Windows, Linux, Mac OS, Solaris, AIX, DEC Unix, FreeBSD, OS/2, Irix, dan sebagainya.
- 3) Sistem otorisasi MySQL yang fleksibel mengizinkan beberapa atau semua *database privileges* (misalnya, *privilege* untuk membuat basis data atau menghapus data) kepada *user* atau sekelompok *user* tertentu.
- 4) MySQL mampu menangani basis data lebih dari 50 juta baris. Batas ukuran *file* untuk sebuah tabel secara *default* adalah 4 GB dan batas ukuran tersebut dapat ditingkatkan hingga 8 juta TB (jika sistem operasi yang digunakan mampu menanganinya).
- 5) Lisensi *open source* GPL mengizinkan *programmer* untuk memodifikasi perangkat lunak MySQL agar dapat disesuaikan dengan kondisi *environment* tertentu.