

Pembuatan Sistem Penilaian Pegawai Berbasis Web bagi Pejabat dan Pelaksana di Lingkungan Universitas Diponegoro

M Fakhri Darmawan, Indra Waspada, Putut Sri Wasito

Jurusan Ilmu Komputer/Informatika, Fakultas Sains dan Matematika, Universitas Diponegoro

fakhriddarmawan@gmail.com

Abstrak

Penilaian prestasi kerja Pegawai Negeri Sipil (PNS) adalah suatu proses penilaian secara sistematis yang dilakukan oleh pejabat penilai terhadap sasaran kerja pegawai dan perilaku kerja PNS. Sistem penilaian prestasi kerja di Universitas Diponegoro dilakukan berdasarkan Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011. Pelaksanaan penilaian pegawai saat ini masih sering terjadinya kesalahan dan format dokumen hasil penilaian yang tidak sesuai. Pembuatan Sistem Penilaian Pegawai Berbasis Web berfungsi untuk memudahkan dalam melakukan penilaian agar sesuai dengan peraturan yang diterapkan dan mengurangi kesalahan dalam penilaian. Model proses pembangun perangkat lunak yang dipakai adalah *Unified Process*. Kerangka kerja yang digunakan yaitu CodeIgniter versi 2.1.2. Penelitian ini menghasilkan Sistem Penilaian Pegawai berbasis web yang dapat memudahkan dalam melakukan penilaian yang sesuai dengan Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011 sehingga dapat digunakan sebagai acuan dalam kenaikan dan penurunan jabatan dan peringkat bagi pejabat dan pelaksana di lingkungan Universitas Diponegoro.

Kata kunci : Sistem Penilaian, *Unified Process*, CodeIgniter

1. Pendahuluan

Penilaian prestasi kerja Pegawai Negeri Sipil (PNS) adalah suatu proses penilaian secara sistematis yang dilakukan oleh pejabat penilai terhadap sasaran kerja pegawai dan perilaku kerja PNS berdasarkan Peraturan Pemerintah Republik Indonesia Nomor 46 Tahun 2011. Penilaian prestasi kerja PNS bertujuan untuk menjamin objektivitas pembinaan PNS yang dilakukan berdasarkan sistem prestasi dan sistem karier yang dititikberatkan pada sistem prestasi kerja [4]. Sistem penilaian prestasi kerja di Universitas Diponegoro dilakukan berdasarkan format dan aturam yang tertera pada Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011 tentang Pedoman Penetapan, Evaluasi, Penilaian, Kenaikan dan Penurunan Jabatan dan Peringkat bagi Pejabat dan Pelaksana di Lingkungan Universitas Diponegoro.

Unit Pelaksana Teknis (UPT) Pusat Komputer Universitas Diponegoro merupakan salah satu unit pelaksana teknis di bidang pengolahan data yang berada dibawah dan bertanggung jawab langsung kepada Rektor, dan pembinaannya dilakukan oleh Pembantu Rektor I. Pengembangan Sistem Penilaian Pegawai

menggunakan data dan arahan yang didapat dari UPT Puskom Universitas Diponegoro. Pelaksanaan penilaian pegawai saat ini memungkinkan terjadinya kesalahan dalam melakukan penilaian, sehingga dibutuhkan ketelitian yang cukup tinggi oleh tim penilai untuk memverifikasi hasil penilaian, selain itu tim penilai harus mengumpulkan hasil penilaian sebelum melakukan verifikasi sehingga membutuhkan waktu yang lama, format dokumen hasil penilaian seringkali tidak sesuai dengan format dalam Peraturan Rektor Universitas Diponegoro, dan dalam menyajikan hasil penilaian yang sebelumnya cukup susah karena hasil penilaian dipegang oleh masing-masing atasan.

Berdasarkan masalah yang dihadapi saat ini maka diperlukan suatu Sistem Penilaian Pegawai yang bisa diakses kapanpun dan dimanapun. Dalam pembuatan suatu sistem diperlukan rencana dan kontrol untuk alasan utama yaitu mampu mengatur kompleksitas sistem yang dibuat. Pada tahun 1998 data perindustrian mengindikasikan bahwa 26 persen proyek perangkat lunak mengalami kegagalan dan 46 persen mengalami pengeluaran biaya lebih dan waktu pengerjaan yang tidak sesuai [7].

Meskipun keberhasilan proyek perangkat lunak meningkat, proyek perangkat lunak masih memiliki tingkat kegagalan yang tinggi, sehingga diperlukan model proses *Unified Process* dalam pembuatan Sistem Penilaian Pegawai serta untuk mempercepat pembuatan Sistem Penilaian Pegawai agar sesuai dengan waktu pengerjaan diperlukan kerangka kerja CodeIgniter. Adapun ruang lingkup pembangunan sistem ini meliputi :

- a. Sistem Penilaian Pegawai dibuat bagi pejabat dan pelaksana di lingkungan Universitas Diponegoro.
- b. Sistem Penilaian Pegawai dibuat berbasis web dengan menggunakan *framework* PHP CodeIgniter 2.1.2 dan basis data MySQL.
- c. Sistem Penilaian Pegawai dibuat menggunakan model proses *Unified Process* dari *workflow requirement* hingga *workflow test*.
- d. Peraturan dan format penilaian pegawai yang digunakan yaitu Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011.

2. Tinjauan Pustaka

Tata cara penilaian dan evaluasi pegawai diatur berdasarkan Peraturan Rektor No. 197/PER/H7/2011 yang berisi tentang Pedoman Penetapan, Evaluasi, Penilaian, Kenaikan dan Penurunan Jabatan dan Peringkat bagi Pejabat dan Pelaksana di lingkungan Universitas Diponegoro. Peraturan yang berisi tentang penilaian dan evaluasi peringkat bagi pelaksana diatur pada pasal 10 hingga pasal 25.

Evaluasi terhadap pelaksana dilakukan setiap 6 (enam) bulan secara periodik, yaitu pada bulan Januari sampai dengan bulan Juni dan bulan Juli sampai dengan bulan Desember. Evaluasi pelaksana dalam peringkatnya diatur atas dasar penilaian 3 (tiga) komponen yang meliputi Pelaksanaan Pekerjaan dengan bobot penilaian 40%, Disiplin Kehadiran dengan bobot penilaian 30%, dan Sikap dan Perilaku terhadap pekerjaan dengan bobot penilaian 30%. Sistem Penilaian Pegawai dibuat berbasis web.

Elemen yang paling dasar dari *World Wide Web* adalah *web client*, *server*, dan jaringan komputer. *Web client* merupakan perangkat lunak yang digunakan untuk mengakses *server* dan menjalankan perangkat lunak berbasis web [5].

Server merupakan perangkat keras yang digunakan untuk menyediakan informasi untuk *client* [5]. Jaringan komputer adalah alat komunikasi antara *client* dan *server* melalui *Transfer Control Protocol/Internet Protocol* (TCP/IP) [5]. TCP/IP merupakan internet protokol yang digunakan untuk komunikasi antara *web client* dan *server*[5]. TCP dan IP merupakan dua protokol yang berbeda. TCP merupakan protokol yang menyediakan fungsi tambahan selain menghubungkan antar komputer termasuk menyediakan *port* untuk berkomunikasi dengan berbagai aplikasi berbeda pada *server* seperti *mail server*, *file download*, dan basis data [5]. IP merupakan protokol utama yang digunakan pada internet. Fungsi IP adalah untuk memindahkan data dari satu komputer ke komputer lainnya [5].

Bahasa pemrograman yang digunakan untuk pembuatan Sistem Penilaian Pegawai adalah PHP dengan menggunakan *framework* CodeIgniter dan basis data MySQL. PHP pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Rasmus membuat sebuah *tool/script* bernama *Personal Home Page* (PHP) untuk dapat mengetahui jumlah *resume online*-nya [11]. PHP kemudian ditulis ulang oleh Zeev Suraski dan Andi Gutsman selaku programmer inti. PHP Parser ditulis ulang dan diintegrasikan dengan menggunakan *Zend scripting engine*, dan mengubah jalan alur operasi PHP, dan semua fitur tersebut dirilis dalam PHP 4 [3].

PHP (akronim dari *PHP Hypertext Preprocessor*) merupakan bahasa pemrograman yang didesain spesial untuk digunakan dalam *web* [9]. Desain struktur dari bahasa pemrograman PHP hampir sama dengan bahasa C. PHP memiliki kemampuan untuk berinteraksi dengan basis data. PHP mendukung berbagai macam basis data yang ada, seperti MySQL, SQL Server, Access, Oracle, dan basis data lainnya.

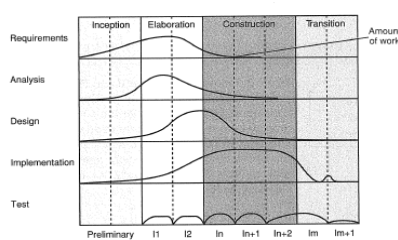
CodeIgniter (CI) adalah sebuah *framework* PHP. *Framework* adalah suatu kerangka kerja yang berupa sekumpulan folder yang memuat *file-file php* yang menyediakan *class libraries*, *helpers*, *plugins* dan *library* lainnya [10].

CI menggunakan konsep Model-View-Controller (MVC). Konsep MVC adalah konsep pemisahan antara logika dengan tampilan dan

basis data. Manfaat dari konsep MVC adalah membuat pengkodean lebih sederhana dan mudah. *Model Script* merupakan kode yang digunakan untuk mengakses data pada basis data. *Model* berisi fungsi di dalam pengolahan basisdata. *View* merupakan pengkodean untuk menampilkan tampilan suatu program. *View* biasanya berupa halaman sederhana menggunakan HTML. *Controller* merupakan pengkodean untuk logika, algoritma dan sebagai penghubung antara model, view, dan sumber lain yang diperlukan untuk mengolah HTTP *request* dan *generate web page*. CI membuat beberapa hal menjadi mudah salah seperti, validasi *form*, operasi *create, read, update* dan *delete* (CRUD) pada basis data [10].

MySQL adalah Sistem Manajemen Basis Data (SMBD) yang sangat populer digunakan bersama PHP. MySQL dapat menyimpan dan mengelola beberapa basis data dalam satu server. Setiap basis data memiliki struktur untuk menyimpan data yang disimpan dalam tabel [9].

Model Proses yang digunakan dalam pembuatan Sistem Penilaian Pegawai yaitu *unified process*. UP merupakan proses pengembangan perangkat lunak secara iterasi yang populer digunakan dengan OOAD [1]. UP bersifat fleksibel, mudah diterapkan, dan menggunakan UML dalam pemodelan perangkat lunak yang dikembangkan. Tahapan dalam UP dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Proses UP

Fase *Inception* merupakan tahapan awal yang singkat untuk memperoleh vision dan batasan sederhana yang dibutuhkan dalam pengembangan perangkat lunak. Fase *Elaboration* merupakan tahapan awal iterasi dalam menginvestigasi, mengimplementasi arsitektur utama perangkat lunak, mengklarifikasi kebutuhan, dan menangani permasalahan yang memiliki resiko

tinggi. Fase *Construction* merupakan tahapan pengembangan sistem hingga sistem siap diuji sebelum diproduksi. Fase *Transition* pengembang berfokus pada perbaikan pemodelan yang ditemukan ketika pengujian beta. Hasil dari tahapan transition merupakan sistem yang siap digunakan, dokumen pendukung dapat diterima, dan mempersiapkan pengguna dalam menggunakan sistem.

Analisis desain pada *unified process* menggunakan OOAD dengan menggunakan pemodelan UML. *Object oriented analysis* berfokus dalam pembuatan deskripsi dari klasifikasi suatu objek. *Object oriented design* berfokus dalam mendefinisikan objek dari perangkat lunak dan mengkolaborasikannya. UML adalah bahasa yang secara umum digunakan dalam membuat suatu business modelling dan notasi digram untuk object oriented modelling. UML digunakan untuk menspesifikasikan, menggambarkan, membangun dan mendokumentasikan OOAD [6].

UML merupakan bahasa pemodelan visual yang digunakan untuk menggambarkan suatu sistem [1]. UML dapat memodelkan suatu perangkat lunak dan sistem lainnya dalam satu interaksi antar objek. Kemampuan UML inilah yang sangat tepat untuk digunakan untuk perangkat lunak berorientasi objek, tidak hanya itu UML juga dapat digunakan untuk membuat bisnis proses dan aplikasi lainnya. Struktur UML terdiri dari *building blocks, common mechanisms*, dan arsitektur.

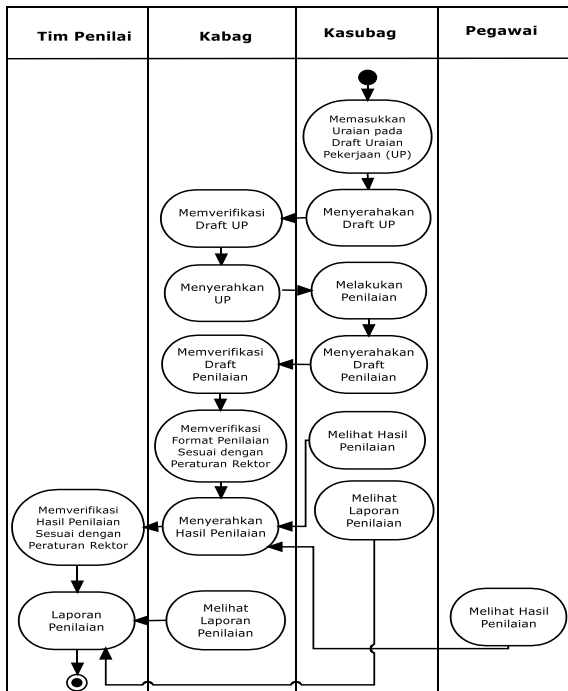
3. Analisis Kebutuhan

Analisis kebutuhan pada fase *inception* yaitu menentukan bisnis model Sistem Penilaian Pegawai sebelum dikembangkan dan bisnis model Sistem Penilaian Pegawai setelah dikembangkan menggunakan pemodelan *activity diagram*. Selain menentukan bisnis model, pada fase *inception* juga didapatkan kebutuhan Sistem Penilaian Pegawai baik itu kebutuhan fungsional maupun non fungsional. Pelaksanaan penilaian pegawai saat ini masih sering memungkinkan terjadinya kesalahan dalam menuliskan formula yang dibuat sehingga dibutuhkan ketelitian yang cukup tinggi oleh tim penilai untuk memverifikasi hasil penilaian, tim penilai harus

mengumpulkan hasil penilaian sebelum melakukan verifikasi, format dokumen hasil penilaian seringkali tidak sesuai dengan format dalam Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011, dan dalam menyajikan hasil penilaian yang sebelumnya cukup susah karena hasil penilaian dipegang oleh masing-masing atasan.

Activity Diagram sistem penilaian pegawai saat ini dapat dilihat pada Gambar 2 dengan contoh penilaian terhadap staf Kepala Sub Bagian (Kasubag). Sistem Penilaian Pegawai yang dikembangkan akan dapat membantu dalam memverifikasi penilaian, membuat dokumen sesuai dengan format pada Peraturan Rektor, dan memudahkan dalam melakukan penilaian.

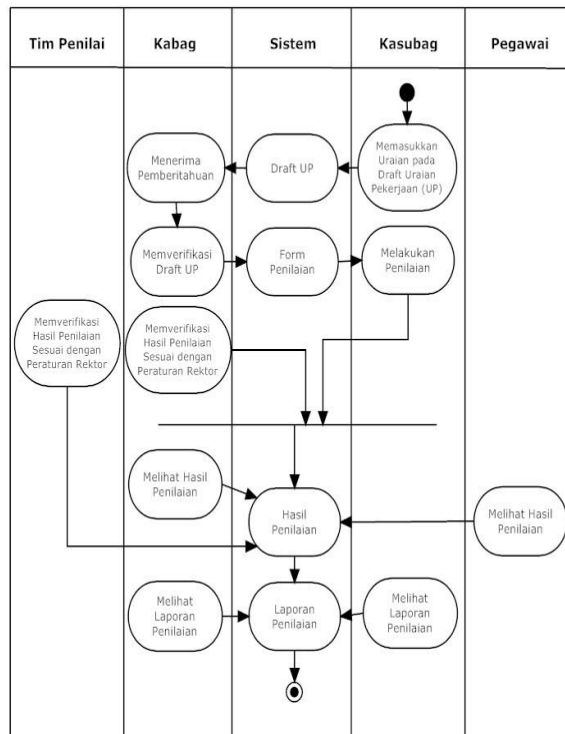
UML Activity Diagram: Sistem Penilaian Pegawai



Gambar 2. *Activity Diagram* Sistem Penilaian Pegawai Sebelum Dikembangkan

Activity Diagram sistem penilaian yang akan dibuat dapat dilihat pada Gambar 3 dengan contoh penilaian terhadap staf Kasubag.

UML Activity Diagram: Sistem Penilaian Pegawai



Gambar 3. Perkiraan *Activity Diagram* Setelah Dikembangkan

Dari analisa terhadap penilaian yang ada saat ini dan peran dari masing-masing pegawai didapat kebutuhan fungsional sebagai berikut :

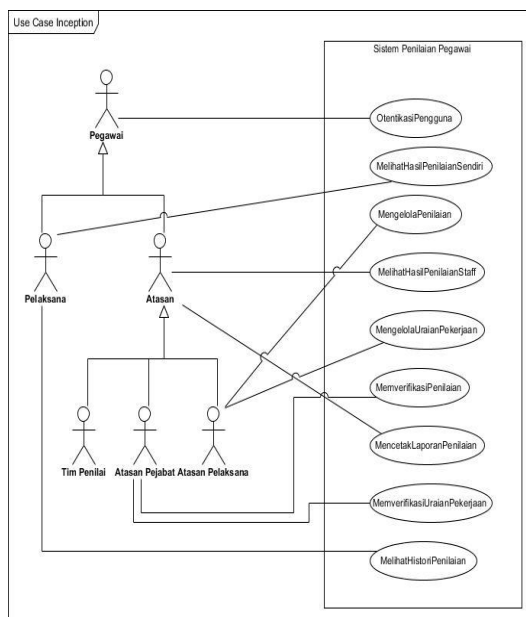
- a. Membutuhkan otentikasi pengguna untuk masuk kedalam sistem
- b. Masing-masing pelaksana dapat melihat hasil penilaian sendiri
- c. Setiap atasan dapat melihat hasil penilaian staff
- d. Atasan pelaksana dapat mengelola penilaian
- e. Atasan pelaksana dapat mengelola uraian pekerjaan
- f. Atasan pejabat dapat memverifikasi penilaian
- g. Atasan pejabat dapat memverifikasi uraian pekerjaan
- h. Pelaksana dapat melihat histori penilaian
- i. Tim penilai dapat mengelola data periode
- j. Tim penilai dapat mengelola data pegawai
- k. Tim penilai dapat melihat dan mencetak hasil penilaian seluruh pegawai
- l. Tim penilai dapat melihat dan mencetak penilaian seluruh pegawai

- m. Tim penilai dapat melihat dan mencetak uraian pekerjaan seluruh pegawai

Kebutuhan non fungsional dari Sistem Penilaian Pegawai yaitu:

- a. Pengguna mengakses Sistem Penilaian Pegawai melalui *Web Browser*
- b. Pengguna mengakses Sistem Penilaian Pegawai membutuhkan koneksi internet

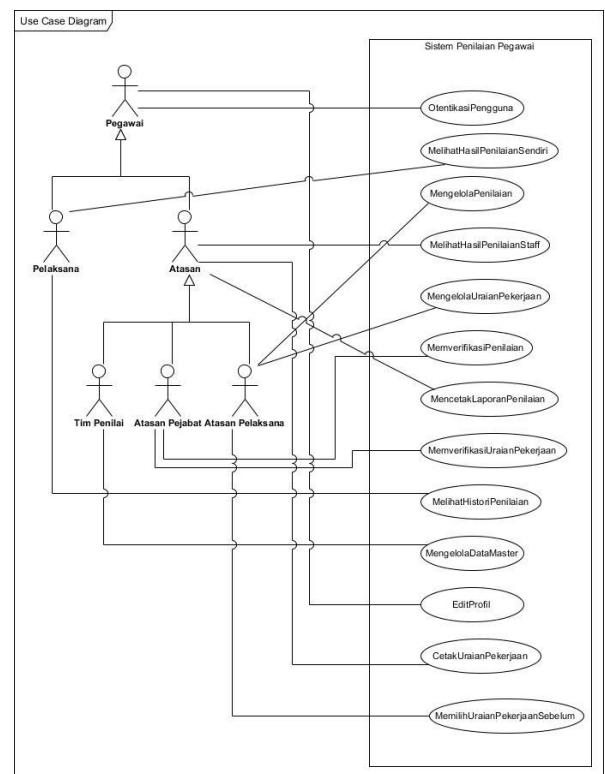
Pemodelan pada *use case* sistem pada fase *Inception* dapat dilihat pada Gambar 4.



Gambar 4. Pemodelan *Use Case* Sistem Penilaian Pegawai

Analisis kebutuhan pada fase *elaboration* berupa perbaikan *use case diagram* pada fase *inception*, dan membuat detail *use case diagram* menggunakan *use case specification* yang berfungsi untuk membuat urutan aktivitas pada *use case*. Data yang digunakan oleh semua pegawai disebut sebagai data master, sehingga diperlukan peran **Tim Penilai** untuk Mengelola Data Master. Semua **Pegawai** memiliki profil masing-masing, sehingga diperlukan fasilitas untuk dapat mengubah profil. *Use case* untuk **Pegawai** ditambahkan dengan Mengubah Profil. Semua **Atasan** membutuhkan uraian pekerjaan yang dapat dicetak sehingga diperlukan *use case* untuk Mencetak Uraian Pekerjaan. Analisis kebutuhan

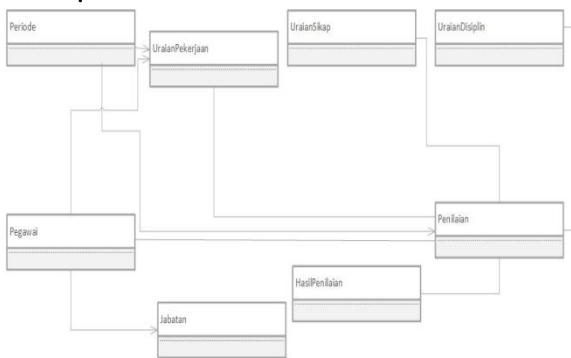
pada fase *construction* berupa memberikan tambahan atau memperbaiki analisis kebutuhan sebelumnya. Kebutuhan sistem pada fase *construction* Sistem Penilaian Pegawai terdapat tambahan yaitu dapat memilih uraian pekerjaan yang sudah dibuat pada periode penilaian sebelumnya. Diagram *use case* Sistem Penilaian Pegawai diubah dengan ditambahkan satu *use case* baru pada aktor **Atasan Pelaksana** yaitu Memilih Uraian Pekerjaan Sebelum, diagram *use case* Sistem Penilaian Pegawai dapat dilihat pada Gambar 5.



Gambar 5.. *Use Case* Fase *Construction*

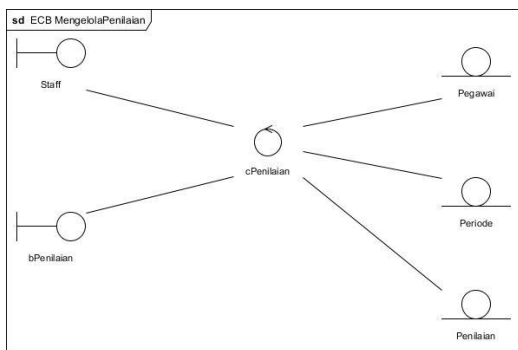
4. Analisis

Analisis pada fase *inception* meliputi pemodelan analisis kelas. Kelas diagram yang dihasilkan pada fase *inception* dapat dilihat pada Gambar 6.



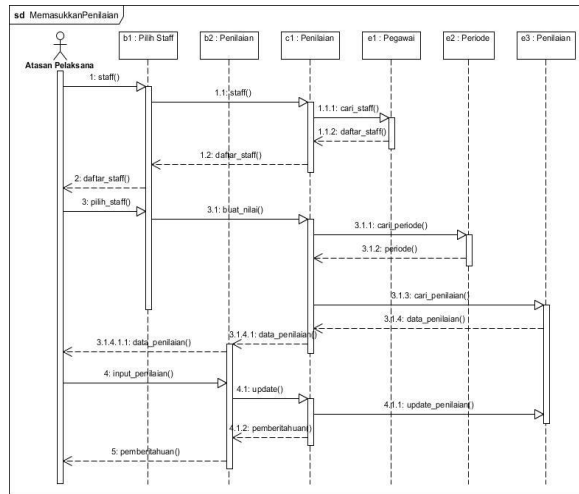
Gambar 6. Kelas Diagram Fase Inception

Setelah kelas diagram sudah terbuat, fase selanjutnya dalam pembuatan *sequence diagram* akan lebih mudah, karena dalam pembuatan *sequence diagram* dapat mengacu pada kelas diagram dan *use case specification* yang telah dibuat. Analisis kelas yang dilakukan pada fase *elaboration* berdasarkan pada kelas diagram dan *use case specification* yang berfungsi untuk menghasilkan arsitektur sistem secara keseluruhan. Ketika mengidentifikasi tingkah laku sistem, dapat menggunakan tiga elemen kunci yaitu *entity*, *control*, dan *boundary*. *ECB pattern* digunakan untuk menentukan sistem *behavior* untuk masing-masing *use case*. Salah satu *ECB pattern* Sistem Penilaian Pegawai yaitu *ECB pattern MengelolaPenilaian*.



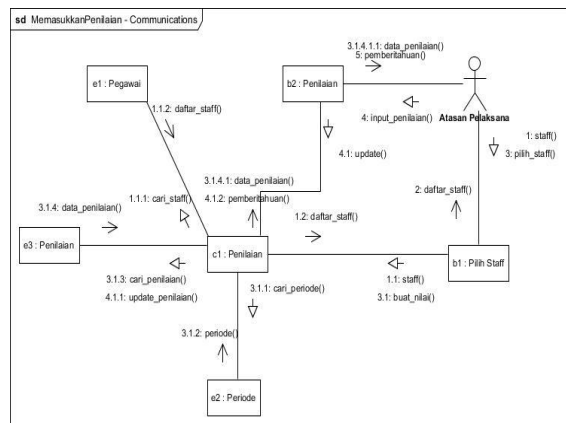
Gambar 7. ECB Pattern MengelolaPenilaian

Dalam pemodelan *sequence diagram* dapat mengacu pada *use case specification* dan kelas diagram pada fase *inception*. *Sequence diagram* untuk MemasukkanPenilaian dapat dilihat pada Gambar 8.



Gambar 8. Sequence Diagram MemasukkanPenilaian

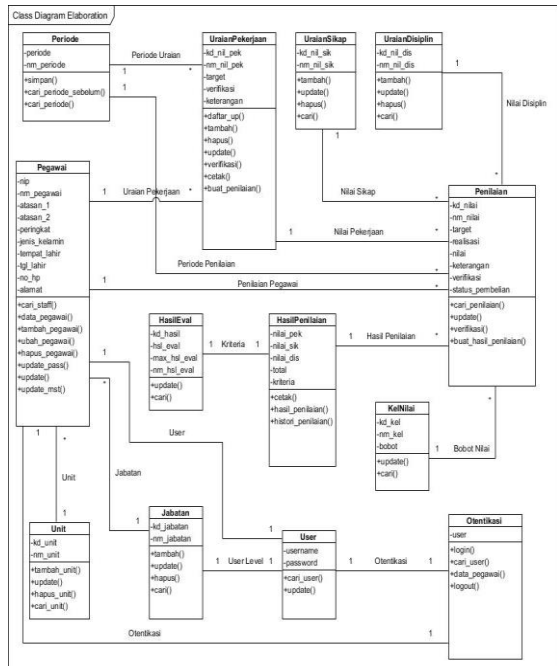
Communication MemasukkanPenilaian dapat dilihat pada Gambar 9.



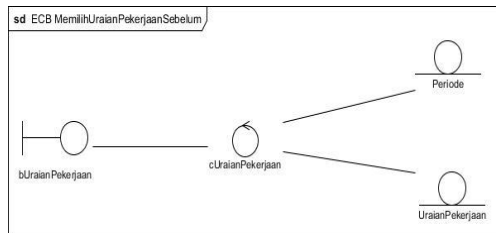
Gambar 9. Communication Diagram MemasukkanPenilaian

Setelah hubungan antar objek didapat menggunakan *communication diagram* maka dihasilkan kelas diagram pada fase *elaboration*, kelas diagram dapat dilihat pada Gambar 10.

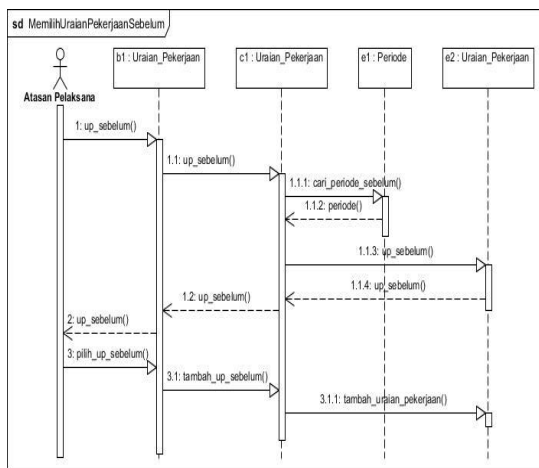
Analisis kelas pada fase *construction* Sistem Penilaian Pegawai yaitu penambahan *use case MemilihUraianPekerjaanSebelum* sehingga kelas diagram Sistem Penilaian Pegawai juga perlu diubah. *ECB Pattern* untuk *use case MemilihUraianPekerjaanSebelum* dapat dilihat pada Gambar 11.



Gambar 10. Analisis Kelas Elaboration

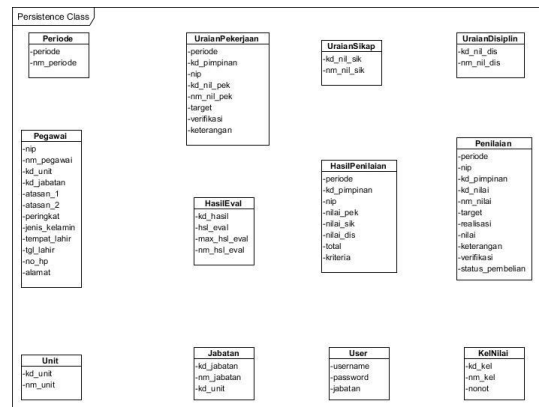


Gambar 11. ECB Pattern MemilihUraianPekerjaanSebelum



Gambar 12. Sequence Diagram MemilihUraianPekerjaanSebelum

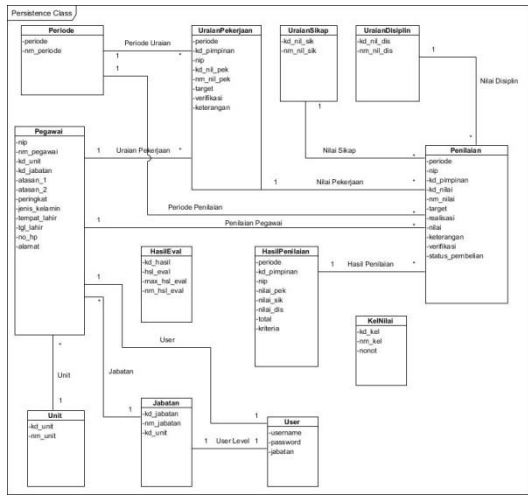
Desain pada *unified process* bertujuan untuk mendapatkan arsitektur sistem yang lengkap sehingga mempermudah dalam membuat implementasi program. Tahapan desain pada fase *inception* hanya membuat tampilan desain antarmuka fungsi utama dari Sistem Penilaian Pegawai yaitu uraian pekerjaan dan penilaian. Desain fase *elaboration* terdapat dua tahap yaitu desain basis data dan desain antarmuka. Desain basis data pada fase elaboration dilakukan sesuai dengan analisis kelas yang telah dibuat. Sebelum membuat desain basis data pertama kali menentukan *persistance class* dari kelas diagram Sistem Penilaian Pegawai.



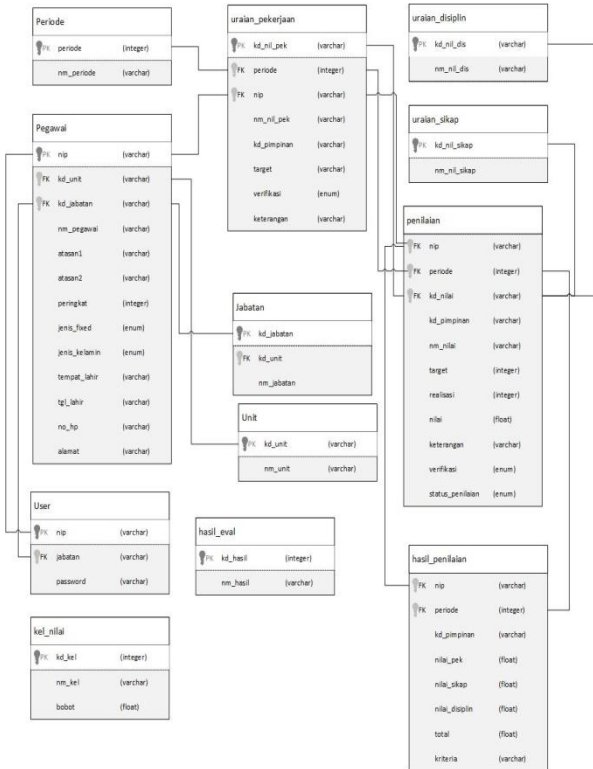
Gambar 13.. Persistence Class Sistem Penilaian Pegawai

Data mapping untuk Sistem Penilaian Pegawai yang pertama kali dilakukan yaitu menghapus operasi pada kelas diagram yang merupakan *persistance class*. Kelas yang memiliki *multiplicity* satu ke banyak maka kunci primer pada kelas yang memiliki *multiplicity* banyak. Kelas yang memiliki hubungan satu ke banyak yaitu kelas **Periode** dengan **UraianPekerjaan**, **Jabatan** dengan **Pegawai**, **Unit** dengan **Pegawai**, **UraianPekerjaan** dengan **Penilaian**, **UraianSikap** dengan **Penilaian**, **UraianDisiplin** dengan **Penilaian**, dan **HasilPenilaian** dengan **Penilaian**. Data mapping dapat dilihat pada Gambar 14. Dari data mapping tersebut didapat struktur tabel dari masing-masing kelas .

5. Desain

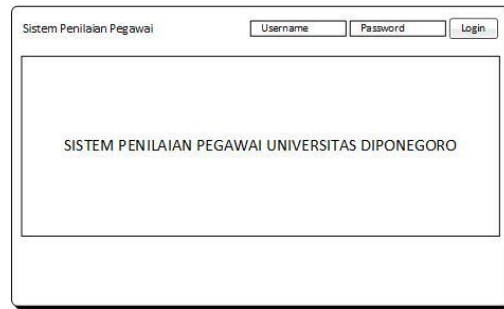


Gambar 14. Data Mapping Sistem Penilaian Pegawai



Gambar 15. Desain Basis Data Sistem Penilaian Pegawai

Desain pada fase *elaboration* tidak hanya membuat desain basis data tetapi juga membuat desain antarmuka Sistem Penilaian Pegawai. Desain halaman awal sistem penilaian pegawai seperti pada Gambar 16.



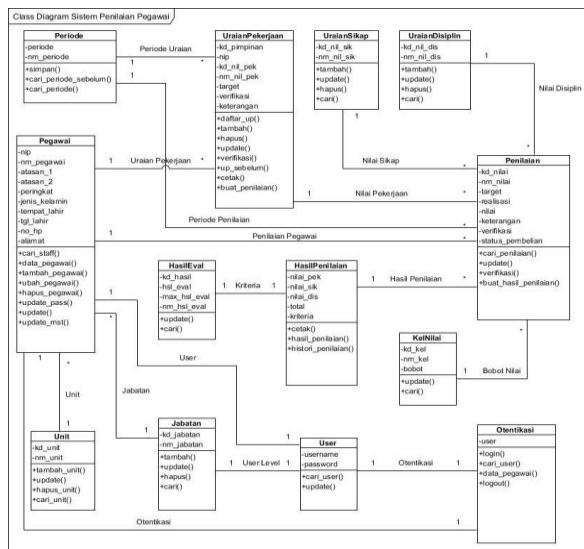
Gambar 16. Desain Halaman Awal Sistem Penilaian Pegawai

Desain antarmuka untuk memilih pegawai yang akan dibuatkan uraian pekerjaan dapat dilihat pada Gambar 17.



Gambar 17. Desain Antarmuka Pilih Pegawai

Desain kelas pada fase *construction* menambahkan memilih uraian pekerjaan sebelum pada analisis kelas yang sudah dibuat. Sehingga kelas diagram Sistem Penilaian Pegawai dapat dilihat pada Gambar 18.

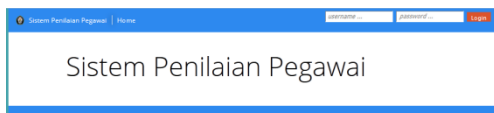


Gambar 18. Kelas Diagram Sistem Penilaian Pegawai

6. Implementasi

Implementasi Pada fase *inception* hanya sebatas pembuatan tampilan awal Sistem Penilaian Pegawai yang berguna untuk mencari tahu bentuk Sistem Penilaian Pegawai yang akan dibuat. Implementasi yang dilakukan hanya menganalisa bagaimana interaksi antara sistem dan pengguna ketika melakukan pembuatan uraian dan memasukkan penilaian.

Tahapan implementasi pada fase *elaboration* dibagi menjadi tiga yaitu implementasi untuk halaman awal, uraian pekerjaan dan penilaian. Implementasi pertama pada fase *elaboration* yaitu membuat halaman awal sistem. Tampilan halaman awal sistem dapat dilihat pada Gambar 19. Halaman awal Sistem Penilaian Pegawai terdapat *form login* untuk masuk pegawai sesuai dengan hak akses masing-masing.



Gambar 19. Halaman Awal Sistem Penilaian Pegawai



Gambar 20. Form *Login* Sistem Penilaian Pegawai

```
function untuk login
public function login()
{
    $this->form_validation->set_rules('password','password','required');
    $this->form_validation->set_rules('username','username','required');
    if($this->form_validation->run()){
        $password = sha1(md5($this->input->post('password')));
        $username = $this->input->post('username');
        $query = $this->general_model->query("
            SELECT p.nip, u.password, p.um_peg, p.kd_unit, p.kd_jabatan, p.atasan
            FROM user u INNER JOIN pegawai p ON p.nip=u.nip
            WHERE u.nip='{$username}' AND u.password='{$password}'
        ");
        $user = $query->row();
        if(empty($user))
            if($user->nip==$username && $user->password==$password){
                $session = array(
                    'nip' => $user->nip,
                    'um_peg' => $user->um_peg,
                    'kd_unit' => $user->kd_unit,
                    'kd_jabatan' => $user->kd_jabatan,
                    'atasan' => $user->atasan,
                    'atasan2' => $user->atasan2,
                    'peringkat' => $user->peringkat,
                    'level' => $this->general_model->get_level($user->kd_jabatan),
                );
            }
        else
            $this->session->sessdata('level')='';
        redirect('home');
    }
}

```

Gambar 21. Fungsi *Login* Pada Kelas Otentikasi

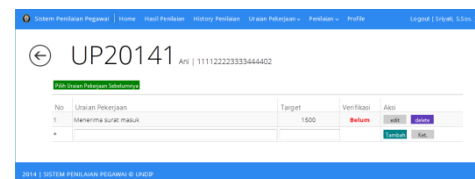
Implementasi selanjutnya yaitu implementasi uraian pekerjaan yang dapat dilihat pada Gambar 22. Tampilan dari uraian pekerjaan dapat dilihat pada Gambar 23. Uraian pekerjaan

yang dimasukkan terdiri dari nama uraian pekerjaan, target, dan keterangan.

```
function untuk menampilkan halaman uraian pekerjaan
public function buat_up()
{
    // check session / login
    if($this->session->sessdata('level')=='') or ($this->session->sessdata('level')=='7'){
        redirect('home');
    }
    // data pegawai
    $nip = $this->uri->segment(1);
    $where = array('nip'=>$nip);
    $pegawai = $this->general_model->get('','pegawai',$where);
    $pegawai = $pegawai->row();
    if(empty($pegawai))
        redirect('uraian_pekerjaan/index');
    // data uraian pekerjaan
    $where_up = array(
        'nip' => $nip,
        'periode' => $this->periode,
    );
    $subdata['profile_pegawai'] = $pegawai;
    $subdata['periode'] = $this->periode;
    $uraian_pekerjaan = $this->general_model->get('','uraian_pekerjaan',$where_up);
    $uraian_pekerjaan = $uraian_pekerjaan->result();
    $subdata['uraian_pekerjaan'] = empty($uraian_pekerjaan)?array():$uraian_pekerjaan;
    // composite view
    $data['content'] = $this->load->view('uraian_pekerjaan/buat_uraian_pekerjaan',$subdata,TRUE);
}

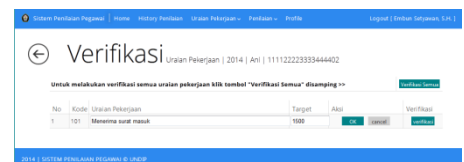
```

Gambar 22. Fungsi Untuk Menampilkan Halaman Uraian Pekerjaan



Gambar 23. Tampilan Uraian Pekerjaan Sistem Penilaian Pekerjaan

Implementasi pada fase *construction* berisi tentang membuat secara keseluruhan berdasarkan kelas diagram pada tahapan sebelumnya. Beberapa implementasi pada fase *construction* yaitu membuat verifikasi uraian pekerjaan dan verifikasi penilaian.



Gambar 24. Tampilan Verifikasi Uraian Pekerjaan

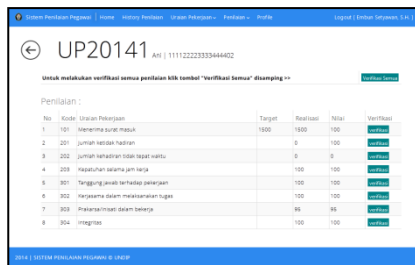
```
function memverifikasi uraian pekerjaan
public function verifikasi_per_element()
{
    if($this->input->is_ajax_request()){
        $nip = $this->uri->segment(1);
        $kd_jabatan = $this->uri->segment(2);
        $subdata['nip'] = $nip;
        // ambil pegawai
        $where = array('nip' => $nip);
        $step = $this->pegawai_model->find('',$where);
        $pegawai = $step->row();
        // cari atasan
        $where = array('kd_jabatan'=>$pegawai->atasan);
        $step = $this->pegawai_model->find('',$where);
        $kd_atasan = $step->row();
        // verifikasi penilaian, set flag verifikasi menjadi 1
        $where = array(
            'periode' => $this->periode,
            'kd_jabatan' => $kd_jabatan,
            'nip' => $nip,
        );
        $update = array_merge(array('flag_verifikasi' => $kd_atasan), $where);
        $this->uraian_pekerjaan_model->update($update,$where);
    }
}

```

Gambar 25. Fungsi Memverifikasi Uraian Pekerjaan

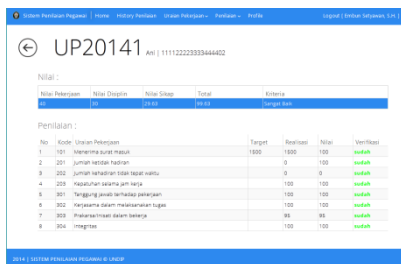
```
function verifikasi_per_nilai()
{
    public function verifikasi_per_element()
    {
        if($this->input->is_ajax_request())
        {
            $nip = $this->uri->segment(1);
            $skd_elem_nilai = $this->uri->segment(2);
            $op = $this->uri->segment(3);
            // ambil pegawai
            $where = array('nip'=>$nip);
            $pegawai = $this->pegawai_model->find($where);
            // verifikasi penilaian, set flag verifikasi menjadi 1
            $where = array(
                'periode' => $this->periode,
                'nip' => $nip,
                'flag_penilaian' => '1',
            );
            $where_update = array_merge(array('ni_nilai' => $skd_elem_nilai, $where);
            $this->penilaian_model->update(array('flag_verifikasi'=>'1'), $where_update);
            // check apakah semua penilaian sudah diverifikasi
            // jika semua sudah diverifikasi semua maka finalkan score
            $where_find = array_merge(array('flag_verifikasi'=>'1'), $where); // merge array
            $show_verifikasi = $this->penilaian_model->find($where_find);
            if(empty($show_verifikasi))
            {
                $skd_elem_nilai = $this->hitung_total_nilai($nip, $this->periode);
            }
        }
    }
}
```

Gambar 26. Fungsi Memverifikasi Penilaian



Gambar 27. Verifikasi Penilaian Pegawai

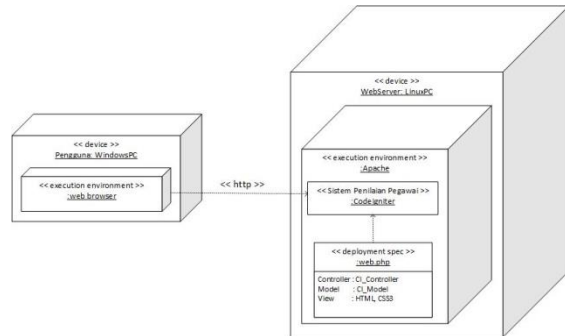
Implementasi pada fase *transition* hanya memperbaiki beberapa fungsi yang salah pada fase *construction* sehingga dihasilkan Sistem Penilaian Pegawai yang telah layak digunakan. Implementasi pada fase *transition* yaitu memperbaiki beberapa kesalahan yaitu penilaian untuk yang sudah diverifikasi tidak dapat dilakukan ubah penilaian lagi, nilai pada hasil penilaian hanya boleh dua desimal, dan hasil evaluasi dapat diisikan nilai desimal. Perbaikan pada penilaian dan hasil evaluasi sudah diujicoba kembali, untuk tampilan penilaian dapat dilihat pada Gambar 28.



Gambar 28. Ubah Penilaian Sudah Diverifikasi

Setelah sistem dibuat secara keseluruhan maka diperlukan suatu *deployment diagram* untuk memudahkan dalam melakukan *deployment*. *Deployment diagram* terdiri dari *node* dan *artifact*. *Node* merupakan representasi dari perangkat keras yang digunakan. *Artifact* merupakan representasi dari perangkat lunak

yang digunakan. *Node* yang digunakan pada Sistem Penilaian Pegawai yaitu PC pengguna dan web server. *Artifact* yang digunakan yaitu CodeIgniter 2.1.2 dan *web browser*. *Deployment diagram* Sistem Penilaian Pegawai dapat dilihat pada Gambar 29.



Gambar 29. Deployment Diagram Sistem Penilaian Pegawai

Pengguna menggunakan windowsPC memerlukan *web browser* untuk dapat mengakses Sistem Penilaian Pegawai yang terletak pada *web server* linux melalui http. Sistem Penilaian Pegawai membutuhkan Apache untuk dieksekusi melalui *web browser* oleh pengguna. Sistem Penilaian Pegawai dibuat menggunakan CodeIgniter dengan spesifikasi controller menggunakan CI_controller, model menggunakan CI_model, dan view menggunakan HTML dan CSS.

7. Pengujian

Tahapan pengujian dilakukan dengan cara pengujian pada masing-masing implementasi ditiap fase pada *unified process*. Pengujian dilakukan secara *blackbox*.

Pengujian pada fase *inception* tidak dilakukan karena implementasi pada fase *inception* hanya menganalisa interaksi yang dilakukan antara sistem dan pengguna.

Pengujian pada fase *elaboration* berfokus pada pengujian *form login*, uraian pekerjaan dan penilaian. Berdasarkan hasil pengujian pada fase *elaboration* fungsi *login*, uraian pekerjaan dan penilaian sudah sesuai dengan yang diharapkan.

Pengujian pada fase *construction* meliputi pengujian verifikasi uraian pekerjaan

memverifikasi penilaian, melihat hasil penilaian sendiri, melihat hasil penilaian staff, mencetak laporan penilaian, mengelola data master, melihat histori penilaian, melihat uraian pekerjaan sebelum. Pengujian fase *construction* terdapat beberapa kesalahan yaitu penilaian untuk yang sudah diverifikasi seharusnya tidak dapat dilakukan ubah penilaian lagi, nilai pada hasil penilaian bernilai lebih dari dua desimal, dan hasil evaluasi tidak dapat diisikan nilai desimal. Kesalahan pada fase *construction* akan diperbaiki pada fase *transition*.

Pengujian pada fase *transition* hanya dilakukan pengujian terhadap beberapa fungsi yang terdapat kesalahan pada saat fase *construction* dan diperbaiki di fase *transition*.

Berdasarkan hasil pengujian hingga fase *transition* Sistem Penilaian Pegawai dianggap oleh UPT Puskom Universitas Diponegoro dapat memudahkan dalam melakukan penilaian yang sesuai dengan Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011.

8. Penutup

Dihasilkan sebuah Sistem Penilaian Pegawai untuk memudahkan UPT Puskom Universitas Diponegoro dalam melakukan penilaian yang sesuai dengan Peraturan Rektor Universitas Diponegoro Nomor 197/PER/H7/2011 sehingga dapat digunakan sebagai acuan dalam kenaikan dan penurunan jabatan dan peringkat bagi pejabat dan pelaksana di lingkungan Universitas Diponegoro. Sistem Penilaian Pegawai ini dikembangkan menggunakan *framework* PHP CodeIgniter yang berbasis Objek dan dapat digunakan di lingkungan Universitas Diponegoro melalui *web browser* didalam jaringan Universitas Diponegoro. Pengembangan sistem informasi ini menggunakan *Unified Process* sehingga memudahkan ketika diperlukan penelusuran dan modifikasi kebutuhan.

Daftar Pustaka

- [1] Arlow, J. & Neustadt, I., 2005. *UML2 and The Unified Process : Practical Object Oriented Analysis and Design*. s.l.:Pearson Education Inc.
- [2] Rektor Universitas Diponegoro, 2011. Peraturan *Rektor Universitas Diponegoro Nomor 197/PER/H/2011*. Semarang: s.n.
- [3] Dwiarta, L., 2010. Menyelam dan Menaklukan Samudra PHP. [Online] Available at: <http://www.ilmuwebsite.com> [Accessed 07 Juni 2012].
- [4] Pemerintah Republik Indonesia., 2011. Peraturan *Pemerintah Republik Indonesia Nomor 46 Tahun 2011*. Jakarta: s.n.
- [5] Jackson, J. C., 2007. *Web Technologies a Computer Science Perspective*. New Jersey: Pearson Prentice Hall.
- [6] Larman, C., 2004. *Applying UML and Pattern*. New Jersey: Prentice Hall.
- [7] Pressman, R. S., 2002. *Software Engineering: a Practitioner's Approach*. New York: McGraw-Hill.
- [8] Rumbaugh, J., Jacobson, I. & Grady, B., 1999. *The Unified Modeling Language Reference Manual*. Reading: Addison Wesley.
- [9] Suehring, S. & Valade, J., 2013. *PHP, MySQL, JavaScript & HTML5 All In One For Dummies*. New Jersey: John Wiley & Sons.
- [10] Upton, D., 2007. *Codeigniter for Rapid PHP Application Development*. Birmingham: Packt Publishing.
- [11] Valade, J., 2010. *PHP & MySQL for Dummies*. 4th ed. Indianapolis: Wiley Publishing Inc.