

PENERAPAN EXTREME PROGRAMMING DALAM MEMBANGUN APLIKASI PENGADUAN LAYANAN PELANGGAN (PaLaPa) PADA PERGURUAN TINGGI

Rezania Agramanisti Azdy¹, Arsia Rini²

¹Program Studi Teknik Informatika, STMIK PalComTech, Palembang

²Program Studi Manajemen Informatika, Politeknik PalComTech, Palembang

Email: ¹rezania@palcomtech.ac.id, ²arsia_rini@palcomtech.ac.id

(Naskah masuk: 19 Februari 2018, diterima untuk diterbitkan: 18 Mei 2018)

Abstrak

Layanan pengaduan yang dimiliki oleh Perguruan Tinggi umumnya hanya menjadikan mahasiswa sebagai fokus utama dalam peningkatan layanan, hal ini dikarenakan mahasiswa adalah pengguna langsung jasa yang ditawarkan oleh Perguruan Tinggi. Layanan pengaduan tidak memfasilitasi pengaduan yang mungkin dapat diberikan oleh pengguna jasa lainnya, seperti pengguna lulusan atau masyarakat sasaran kegiatan pengabdian masyarakat yang dilakukan oleh dosen Perguruan Tinggi tersebut. Penelitian ini bertujuan membangun sebuah aplikasi berbasis web yang dapat digunakan seluruh pengguna jasa Perguruan Tinggi untuk melakukan pengaduan. Pembangunan aplikasi menerapkan metodologi rekayasa perangkat lunak agar dapat menghasilkan perangkat lunak tepat waktu. Metodologi yang digunakan adalah *Extreme Programming*. Hasil dari penelitian menunjukkan bahwa penerapan *Extreme Programming* pada pengembangan aplikasi mengakibatkan pengembangan dapat dilakukan secara cepat dengan jumlah anggota tim yang minimal. Pengembangan dapat dilakukan dengan cepat karena pengembangan aplikasi dimulai dengan membangun komponen yang paling sederhana terlebih dahulu, yaitu komponen verifikasi akun, yang dilanjutkan dengan komponen lainnya. Hasil pengujian menggunakan metode *black box* aplikasi dapat dijalankan sebagaimana mestinya dan memenuhi *acceptance criteria* pada *user story*.

Kata kunci: layanan pengaduan, tindakan, rekayasa perangkat lunak

IMPLEMENTATION OF EXTREME PROGRAMMING IN BUILDING APPLICATION FOR CUSTOMER SERVICE COMPLAINTS (PaLaPa) ON COLLEGE

Abstract

The complaint service system owned by College generally only focuses on the students in the improvement of services, this is because students are the direct users of the services offered by the College. Complaints service did not facilitate the complaints which may be given by other service users, such as graduates' user or community targeted in the society devotion activities undertaken by the College lecturer. This research aims to build a web-based application that can be used throughout the service users of the College to conduct complaints. Application development applying software engineering methodologies in order to provide the software on time. The methodology used is *Extreme Programming*. The result of the research shows that the application of the *Extreme Programming* in the development of applications resulted in development can be done quickly with a minimal number of the team member. The development can be done quickly because the application development started by building the simplest component in advance, i.e. account verification component, followed by other components. The results of the testing using *black box* method was the application can be run properly and meets the *acceptance criteria* on a *user story*.

Keywords: complaint services, actions, software engineering

1. PENDAHULUAN

Perguruan tinggi (PT) adalah sebuah lembaga atau institusi penyelenggara pendidikan tinggi, baik yang diselenggarakan oleh pemerintah maupun oleh masyarakat (swasta). Menurut UU Nomor 12 Tahun 2012 pasal 58, Perguruan Tinggi memiliki fungsi dan peran sebagai: 1) wadah pembelajaran mahasiswa

dan masyarakat, 2) wadah pendidikan calon pemimpin bangsa, 3) pusat pengembangan Ilmu Pengetahuan dan Teknologi, 4) pusat kajian kebajikan dan kekuatan moral untuk mencari dan menemukan kebenaran; dan 5) pusat pengembangan peradaban bangsa, yang kesemuanya dilaksanakan melalui kegiatan Tridharma yang telah ditetapkan

dalam statuta PT. Dari fungsi dan peran yang telah ditetapkan dalam UU diatas dan kegiatan Tridharma pada PT, dapat dilihat bahwa PT merupakan sebuah lembaga yang melayani masyarakat dalam bidang Pendidikan Tinggi. Masyarakat disini bukan hanya mahasiswa sebagai sasaran utama kegiatan Pendidikan Tinggi saja, melainkan juga masyarakat umum yang dapat dijadikan sasaran bagi PT dalam transfer ilmu pengetahuan dan teknologi yang dimiliki oleh sivitas akademika PT khususnya dosen. Selain itu, dalam menjamin keberlangsungan sebuah Perguruan Tinggi, kompetensi lulusan yang dihasilkan harus sesuai dengan kebutuhan perusahaan maupun organisasi pengguna keahlian lulusan. Sehingga secara tidak langsung PT dapat dikatakan sebagai sebuah tempat produksi sumber daya manusia kompeten yang nantinya akan digunakan oleh perusahaan baik milik negara, maupun swasta.

Dalam menjalankan fungsi dan perannya, setiap PT berusaha memberikan pelayanan prima kepada pengguna jasanya. Pengguna jasa pada PT dapat dikategorikan menjadi 2, yaitu pengguna langsung maupun pengguna tidak langsung. Pengguna langsung adalah mahasiswa PT itu sendiri dan pengguna tidak langsung adalah pengguna lulusan maupun masyarakat yang menjadi sasaran kegiatan pengabdian PT. Pelayanan yang diberikan adalah pelayanan terbaik tidak hanya menurut PT sendiri melainkan juga apa yang dirasakan dari sisi pelanggan. Beberapa cara telah dilakukan PT dalam menampung keluhan maupun kritik dan saran dari pelanggan khususnya mahasiswa, baik dengan mengirimkan pesan melalui media tertentu (Gambar 1) atau dengan membangun aplikasi yang umumnya berbasis web yang dapat diakses oleh mahasiswa secara *online* seperti yang dituliskan oleh Bakri (2016) maupun pada penelitian yang dilakukan oleh Ferdiansyah, Jasri & Widjianto (2016).

Aplikasi layanan pengaduan yang diberikan oleh PT umumnya hanya berfokus terhadap pelayanan yang diberikan kepada mahasiswa sebagai pengguna langsung terhadap jasa yang ditawarkan PT. Aplikasi tidak memberikan fasilitas terhadap pengaduan yang diberikan oleh pengguna jasa lainnya, misalnya oleh pengguna lulusan yang ingin memberikan pengaduan mengenai kompetensi lulusan yang dihasilkan agar dapat dijadikan masukan dalam memperbaiki kualitas mahasiswa yang dihasilkan maupun oleh masyarakat sasaran pengabdian masyarakat yang ingin memberikan saran mengenai materi pengabdian yang tidak sesuai dengan proposal kegiatan yang ditawarkan. Penelitian ini bertujuan membangun sebuah aplikasi berbasis web yang dapat digunakan seluruh pengguna jasa PT untuk melakukan pengaduan. Aplikasi nantinya dapat digunakan baik oleh mahasiswa itu sendiri, orang tua, pengguna lulusan, hingga masyarakat umum yang berkaitan dengan kegiatan usaha PT.



Gambar 1. Layanan Konsultasi dan Pengaduan Mahasiswa pada STMIK-Politeknik Palcomtech Melalui Telegram

Pembangunan aplikasi menerapkan metodologi pengembangan perangkat lunak agar dapat menghasilkan perangkat lunak tepat waktu, sesuai dengan analisa kebutuhan sebagai solusi dari permasalahan yang ditemui, dan pemeliharaan serta pengembangan perangkat lunak yang mudah karena adanya dokumentasi yang rinci mengenai sistem yang telah dibangun dan mengapa sistem dibangun.

Salah satu metodologi pengembangan perangkat lunak yang dapat digunakan adalah *Extreme Programming (XP)*. Menurut Beck dalam Prabowo, Sholih & Muqtadiroh (2013), *Extreme Programming* cenderung menggunakan pendekatan berorientasi objek dan sesuai digunakan ketika tim dihadapkan dengan *requirement* yang tidak jelas maupun jika terjadi perubahan *requirement* yang sangat cepat. Penerapan XP pada penelitian yang telah dilakukan oleh Adelin & Effendi (2017), dapat menghasilkan aplikasi dalam kurun waktu lebih cepat dengan jumlah anggota tim yang sedikit. Metodologi XP berfokus pada pengkodean dan sangat bergantung pada ketrampilan pengembang (programmer) (Aitken & Ilango, 2013).

Penelitian mengenai pembangunan aplikasi pengaduan pelanggan pernah dilakukan oleh Gunawan & Agustian (2014). Penelitian ini membahas pembangunan aplikasi pengaduan gangguan listrik di PLN Rayon Banjaran. Penelitian menggunakan System Development Life Cycle (SDLC) untuk pengembangan sistemnya dan menghasilkan aplikasi pelayanan gangguan listrik yang dapat membantu PLN Rayon Banjaran dalam mengolah data perbaikan gangguan listrik atas complain yang diajukan pelanggan melalui sistem secara online.

Penelitian lainnya mengenai aplikasi pengaduan pelanggan pernah dilakukan oleh Masya, Elvina & Simanjuntak (2012) yang membahas rancang bangun sistem pelayanan pengaduan masyarakat pada Divisi Humas Polri berbasis web dengan perancangan sistem dimodelkan menggunakan *Unified Modeling Language (UML)* dan pembangunan sistem menggunakan bahasa pemrograman PHP dan database MySQL. Pengembangan sistem yang digunakan pada penelitian ini adalah metode waterfall dan menghasilkan website sebagai sistem pelayanan publik untuk penyampaian informasi

kepada masyarakat dan menampung pengaduan masyarakat.

Makalah ini membahas penerapan *Extreme Programming* dalam membangun aplikasi pengaduan layanan pelanggan (PaLaPa) yang digunakan pada Perguruan Tinggi. Pada Bab 2 akan dibahas metode penelitian yang dilakukan, Bab 3 menyajikan hasil dan pembahasan dari penelitian yang telah dilakukan, Bab 4 berisi kesimpulan, dan Bab 5 berisi saran mengenai aplikasi yang telah dibangun.

2. METODE PENELITIAN

2.1. Jenis Penelitian

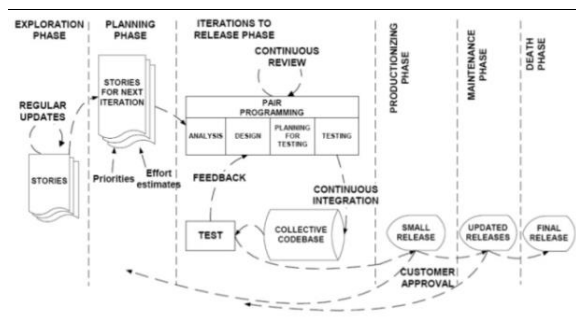
Jenis penelitian yang dilakukan adalah Research and Development (R&D), yaitu penelitian dengan menerapkan langkah-langkah yang ada untuk menghasilkan sebuah produk perangkat lunak (PaLaPa).

2.2. Teknik Pengumpulan Data

Teknik pengumpulan data pada penelitian ini menggunakan metode observasi dan wawancara. Pada metode observasi diperoleh data berupa prosedur dan mekanisme yang ada dalam menerima pengaduan pada studi kasus STMIK Palcomtech, dan pada metode wawancara diperoleh kebutuhan pengguna terhadap aplikasi yang akan dikembangkan. Responden berjumlah 50 orang yang terdiri dari civitas akademika dan pengguna lulusan.

2.3. Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan adalah metode *Extreme Programming*. Abrahamsson dalam Dullemond & Gameren (2009) mendefinisikan proses-proses pada *Extreme Programming* yang digunakan pada penelitian ini seperti diperlihatkan pada Gambar 2.



Gambar 2. Tahapan Siklus Hidup pada XP (Dullemond & Gameren, 2009)

Penjelasan tahapan-tahapan yang ada pada Gambar 2 menurut Dullemond & Gameren (2009) adalah:

1. Exploration

Pada tahapan ini, klien mendefinisikan feature yang mereka harapkan ada pada sistem yang akan dibuat dan di saat yang bersamaan tim pengembang menyesuaikan dengan tools, teknologi, dan praktik-praktik yang akan digunakan pada proyek.

2. Planning

Pada tahapan ini *requirement* yang telah ditentukan pada tahap sebelumnya dipecah menjadi *user stories* dan ditetapkan prioritas pengerjaannya untuk kemudian disepakati konten yang akan dibuat pada release awal (small release).

3. Iterations to Release

Tahapan ini terdiri dari beberapa iterasi untuk dapat menghasilkan sistem pada release awal. Jadwal yang telah ditentukan pada tahapan planning dipecah menjadi beberapa iterasi dengan durasi setiap iterasi berkisar dari satu hingga empat minggu. Iterasi awal difokuskan pada pembangunan arsitektur sistem sebagai pondasi sehingga akan dipilih *requirement* yang memberlakukan pembangunan struktur untuk keseluruhan sistem. *requirement* yang akan dikerjakan pada setiap iterasi dipilih oleh pelanggan. Pada akhir setiap iterasi tes fungsional dilakukan dan di akhir iterasi sistem telah siap untuk diproduksi.

Pembangunan sistem pada tahapan ini dilakukan dengan menerapkan pair programming. Beberapa literatur mendefinisikan pair programming sebagai pendekatan dimana dua orang programmer melakukan pengkodean untuk task yang sama dengan menggunakan satu komputer dan satu keyboard. Satu programmer bertugas sebagai “driver” yang berfokus melakukan pengkodean, dan programmer lainnya bertugas sebagai “navigator” yang melakukan observasi dan mengomentari kode program yang telah ditulis dan mencari alternatif pengkodean lain jika diperlukan (Hannay et al, 2010).

4. Productionizing

Pada tahap ini pengujian ekstra dan pemeriksaan kinerja sistem dilakukan. Dari kegiatan tersebut terdapat kemungkinan ditemukannya perubahan sistem, dan kesepakatan harus dibuat mengenai perlu tidaknya perubahan tersebut diikutkan pada small release atau disesuaikan untuk release berikutnya. Pada tahap ini lamanya iterasi dipersingkat menjadi satu minggu.

5. Maintenance

Pada tahap ini dilakukan telah dilakukan release sistem untuk digunakan oleh pengguna. Sistem harus tetap berjalan dan pada saat yang bersamaan dihasilkan iterasi baru. Untuk dapat melakukan hal itu, maka diperlukan support task dari pihak klien sehingga dapat menghambat kecepatan pengembangan atau diperlukan menambahkan anggota baru ke tim.

6. Death

Ketika seluruh *requirement* pelanggan telah dipenuhi baik yang berkaitan dengan kinerja maupun kehandalan sistem, maka tahapan ini tercapai. Pada tahapan ini ditulis dokumentasi sistem yang diperlukan karena tidak ada lagi perubahan baik pada arsitektur, desain, maupun pengkodean. Tahapan death juga dapat tercapai ketika sistem tidak dapat memberikan hasil sesuai *requirement* yang dibuat atau biaya yang dikeluarkan untuk pengembangan lebih lanjut menjadi terlalu mahal.

Penelitian mengenai pengembangan perangkat lunak menggunakan *Extreme Programming* pernah dilakukan oleh Adelin & Effendi (2017). Penelitian ini membahas pembangunan sistem Audit Mutu Akademik Internal (AMAI). Penelitian ini menghasilkan sebuah sistem yang dapat mengintegrasikan data yang dimiliki oleh klien dengan dokumen mutu akademik yang diperlukan untuk keperluan audit akademik.

Penelitian lainnya pernah dilakukan oleh Ramadhani, Adam & Adi (2015). yang membahas pembangunan aplikasi web distribusi kelompok tani dengan menggunakan metode *Extreme Programming*. Hasil dari penelitian ini adalah aplikasi website yang dapat membantu Kelompok Tani KATATA dalam pengelolaan pemesanan, pengiriman, serta pengelolaan barang.

3. HASIL DAN PEMBAHASAN

Pada tahapan awal pengembangan perangkat lunak, pengguna menentukan *requirement*-nya dan ditulis dalam bentuk *user story*. Menurut Choudhari & Suman (2012)., *user story card* yang baik menerapkan model INVEST yaitu *Independent*, *Negotiable*, *Valuable*, *Estimable*, dan *Small*.

Independent berarti tidak adanya ketergantungan antar *user story*, *Negotiable* berarti seluruh detail diperoleh melalui kolaborasi, *Valuable* berarti memiliki nilai bagi pengguna, *Estimable* berarti dapat diperkirakan baik dalam segi waktu dan biaya, *Testable* berarti dapat ditentukan *acceptance criteria*-nya. Sebuah *user story* dapat dikatakan “selesai” jika *acceptance criteria user story* tersebut telah dipenuhi.

Pembahasan penelitian ini dimulai dengan merangkum *user story* dalam bentuk tabel (Tabel 1). Secara umum *user story* terdiri dari tiga bagian, yaitu judul, deskripsi, dan *acceptance criteria* (Choudhari & Suman, 2012). *User story* pada Tabel 1 dikelompokkan berdasarkan pengguna aplikasi baik secara umum maupun lebih detail berdasarkan kebutuhannya. Pengguna aplikasi adalah pembuat pengaduan yang disebut dengan *complainer*, pihak atau divisi yang dikomplain yang disebut dengan *complainee*, dan admin. *Complainer* dapat berupa mahasiswa, orangtua/wali mahasiswa, perusahaan tempat mahasiswa PKL, dan pihak lainnya yang menggunakan jasa PT. *Complainee* dapat berupa perwakilan dari masing-masing divisi yang terdapat pada PT.

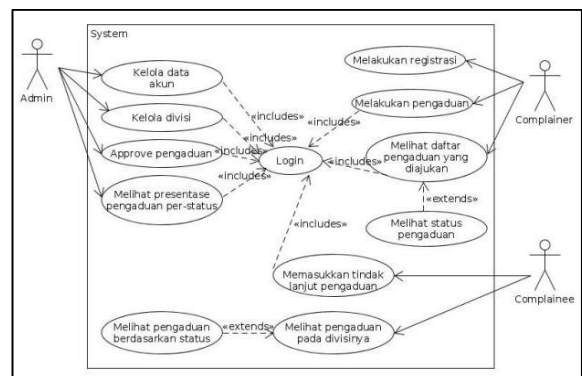
Tabel 1. Rangkuman *user story Card*

No	Judul	Deskripsi	Acceptance Criteria
1	Registrasi	Sebagai pengguna, saya ingin dapat mendaftarkan diri saya agar dapat memiliki akses terhadap aplikasi.	Terdapat menu untuk registrasi agar pengguna dapat memperoleh akun. Hal yang diperlukan dalam pembuatan akun pengguna adalah nama lengkap, <i>username</i> , password, alamat, no handphone, instansi, dan keterangan.
2	Validasi	Sebagai pengguna, saya ingin menggunakan aplikasi hanya jika saya memiliki akses terhadap aplikasi sehingga saya dapat menggunakan aplikasi sesuai dengan hak akses saya.	Tampilan awal aplikasi adalah penjelasan singkat tentang aplikasi. Pengguna dapat menggunakan feature aplikasi jika pengguna memiliki akses terhadap aplikasi, yaitu memiliki <i>username</i> dan password yang telah terdaftar pada sistem.
3	Melakukan pengaduan.	Sebagai <i>complainer</i> , saya ingin melakukan pengaduan yang secara spesifik ditujukan pada divisi tertentu sehingga pengaduan tepat sasaran dan mudah ditangani.	Pada form pengaduan, terdapat komponen yang berisi daftar divisi yang terdapat pada PT. Terdapat penjelasan mengenai masing-masing divisi agar <i>complainer</i> mengetahui divisi mana yang dijadikan sasaran.
4	Memeriksa status pengaduan.	Sebagai <i>complainer</i> , saya ingin melihat status pengaduan yang saya lakukan sehingga saya dapat mengetahui progres pengaduan .	Pada dashboard <i>complainer</i> , terdapat daftar pengaduan yang pernah dilakukan oleh <i>complainer</i> beserta keterangan status setiap pengaduan.
5	Meneruskan pengaduan ke divisi terkait.	Sebagai admin, saya ingin memeriksa terlebih dahulu pengaduan yang telah masuk dan	Pada setiap pengaduan yang masuk, terdapat komponen checklist yang dapat diaktifkan admin sebagai

		meneruskannya ke divisi terkait jika pengaduan tersebut valid sehingga dapat meminimalisasi spam yang masuk.	penanda bahwa pengaduan tersebut valid. Setiap pengaduan yang valid secara otomatis akan diteruskan ke divisi terkait dan dapat dilihat pada daftar pengaduan di divisi tersebut.
6	Daftar pengaduan yang valid dapat dikategorikan berdasarkan divisinya.	Sebagai admin, saya ingin melihat daftar pengaduan yang valid berdasarkan divisi sehingga dapat melihat data pengaduan yang masuk pada divisi tertentu.	Pada halaman data pengaduan yang valid terdapat komponen untuk memasukkan nama divisi. Jika nama divisi telah diberikan, maka data pengaduan yang ditampilkan adalah pengaduan yang sesuai berdasarkan nama divisi.
7	Persentase pengaduan per-status (selesai/sedang diproses/pending).	Sebagai admin, saya ingin melihat persentase pengaduan yang dikelompokkan berdasarkan statusnya sehingga dapat mengetahui banyaknya porsi pengaduan tiap status.	Terdapat sebuah halaman yang menampilkan presentase pengaduan berdasarkan status. Presentase dapat digambarkan menggunakan diagram lingkaran dengan warna yang mewakili masing-masing status.
8	Mengelola divisi.	Sebagai admin, saya ingin mengelola data divisi, sehingga dapat menambahkan jika ada divisi baru, atau menghapus jika ada divisi yang dihilangkan.	Terdapat menu kelola divisi dimana admin dapat menambah, mengedit, dan menghapus divisi.
9	Daftar pengaduan pada setiap divisi dapat dikategorikan berdasarkan statusnya.	Sebagai <i>complainee</i> , saya ingin mengkategorikan daftar pengaduan yang diajukan ke divisi berdasarkan statusnya sehingga dapat diketahui status pengaduan baik yang telah selesai maupun belum.	Pada daftar pengaduan per-divisi, terdapat komponen yang dapat digunakan untuk memilih status. Daftar pengaduan untuk divisi tersebut akan ditampilkan secara keseluruhan terkecuali jika status diaktifkan makaftar pengaduan yang ditampilkan sesuai dengan status yang diminta.
10	Keterangan tindak lanjut terhadap pengaduan yang diberikan.	Sebagai <i>complainee</i> , saya ingin menyimpan data tindak lanjut yang saya lakukan pada setiap pengaduan sehingga dapat dijadikan log tindakan untuk penanganan pengaduan tertentu.	Pada setiap pengaduan per-divisi, terdapat detil lebih mengenai pengaduan termasuk tindak lanjut yang pernah dilakukan terhadap pengaduan tersebut. Tindak lanjut dapat ditambahkan dan diedit tapi tidak dapat dihapus.

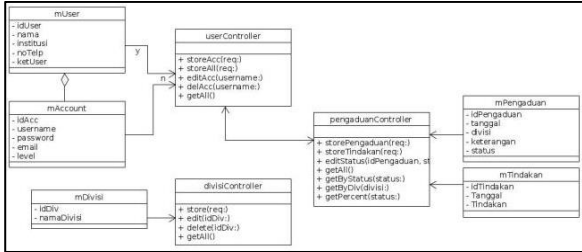
Pada tahapan akhir pengembangan perangkat lunak, dibuat dokumen rancangan sistem menggunakan diagram *Unified Modeling Language* untuk memodelkan perilaku dan struktur statis dari aplikasi dan *Entity Relationship Diagram* untuk memodelkan rancangan *database* yang digunakan.

Diagram *use case* menggambarkan kebutuhan fungsional aplikasi PaLaPa. Pengguna aplikasi digambarkan sebagai aktor-aktor yang berasosiasi dengan *use case* sebagai simbol kebutuhan fungsional yang dapat dilakukan setiap aktor. Pada Gambar 3, pengguna aplikasi berupa admin, *complainer*, dan *complainee*.



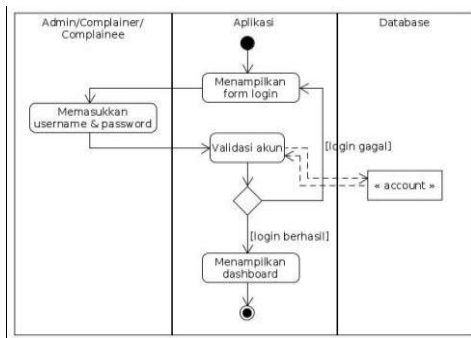
Gambar 3. Use Case Diagram Aplikasi PaLaPa

Struktur statis digambarkan menggunakan *class diagram* seperti diperlihatkan pada Gambar 4. Aplikasi dikembangkan menggunakan pendekatan berbasis objek dengan menggunakan konsep *Model View Controller*.



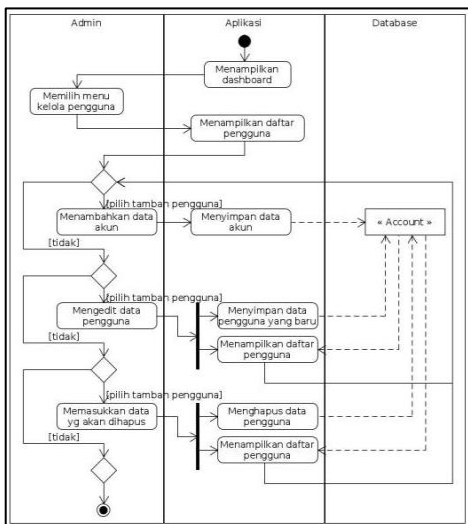
Gambar 4. Class Diagram Aplikasi PaLaPa

Perilaku sistem digambarkan menggunakan *activity diagram* yang dibuat berdasarkan *use case diagram*-nya. *Activity diagram* untuk *use case Log in* menggambarkan aktivitas yang dikerjakan ketika pengguna melakukan *log in* ke dalam aplikasi (Gambar 5). *Activity* ini dilakukan oleh pengguna agar dapat masuk ke dalam aplikasi.



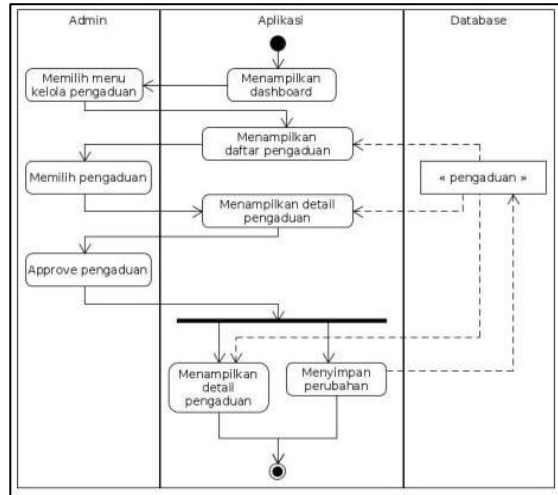
Gambar 5. Activity Diagram untuk Use Case Log in

Activity diagram berikutnya dikategorikan berdasarkan pengguna (aktor) dan *use case*-nya masing-masing. *Activity diagram* untuk aktivitas admin dalam mengelola data akun diperlihatkan pada Gambar 6.



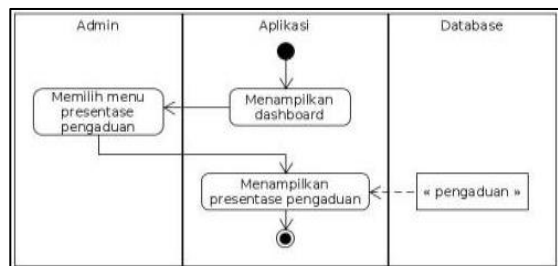
Gambar 6. Activity Diagram Kelola Data Akun

Rangkaian kegiatan dalam aplikasi untuk kelola data divisi serupa dengan alur kegiatan untuk kelola data akun. *Activity diagram* untuk menyetujui pengaduan yang telah masuk diperlihatkan pada Gambar 7.



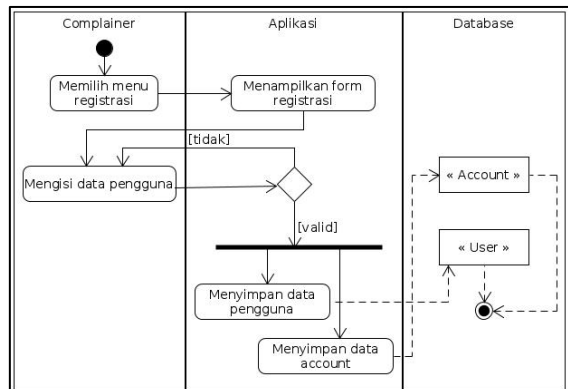
Gambar 7. Activity Diagram Approve Pengaduan

Activity diagram berikutnya memperlihatkan rangkaian proses yang dilakukan aplikasi agar admin dapat melihat pengaduan berdasarkan statusnya (Gambar 8).



Gambar 8. Activity Diagram untuk Melihat Presentase Pengaduan

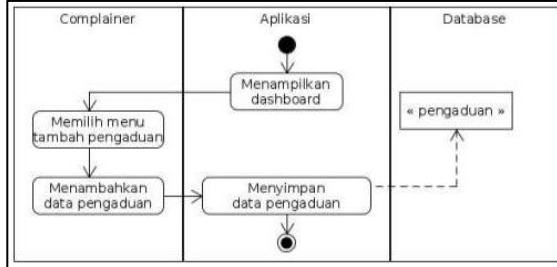
Activity diagram berikutnya memperlihatkan rangkaian kegiatan yang dilakukan aplikasi ketika *complainer* dalam menggunakan aplikasi.



Gambar 9. Activity Diagram untuk Registrasi

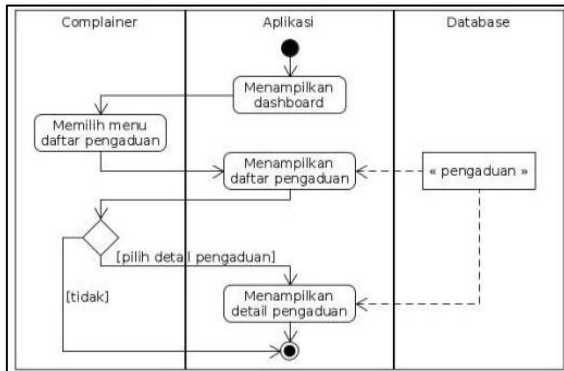
Untuk dapat melakukan pengaduan, *complainer* diharuskan memiliki akun terlebih dahulu dengan cara registrasi, aktivitas yang dilakukan

aplikasi dalam menerima registrasi *complainer* diperlihatkan pada Gambar 9. Setelah memiliki akun, *complainer* dapat melakukan pengaduan dengan aktivitas yang dilakukan aplikasi ditunjukkan pada Gambar 10.



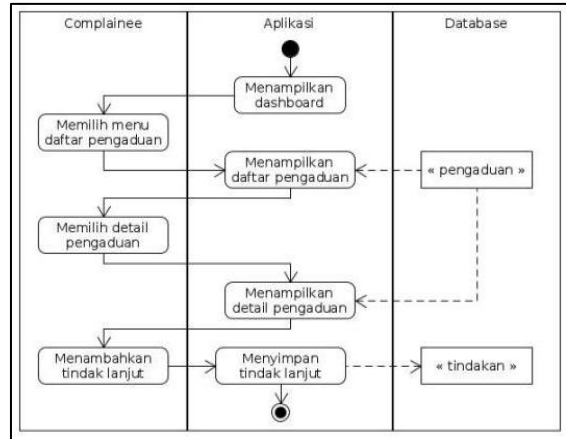
Gambar 10. Activity Diagram untuk Melakukan Pengaduan

Aktivitas aplikasi ketika *complainer* melihat daftar pengaduan yang pernah dilakukan dan melihat masing-masing status pengaduan diperlihatkan pada Gambar 11.



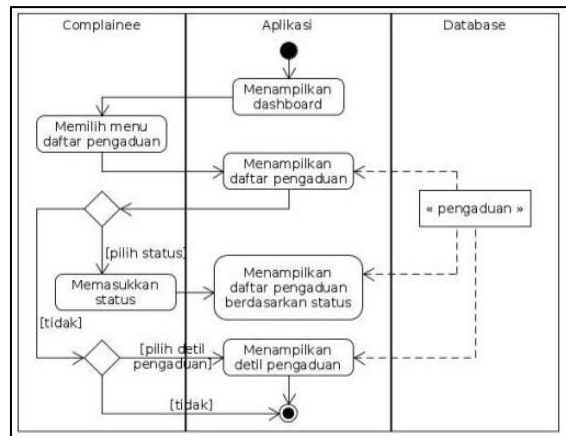
Gambar 11. Activity Diagram untuk Melihat Pengaduan dan Status

Activity diagram berikutnya adalah rangkaian kegiatan yang dilakukan aplikasi ketika *complainee* menggunakan aplikasi. Gambar 12 memperlihatkan rangkaian aktivitas aplikasi ketika *complainee* memasukkan tindak lanjut pengaduan pada divisinya.



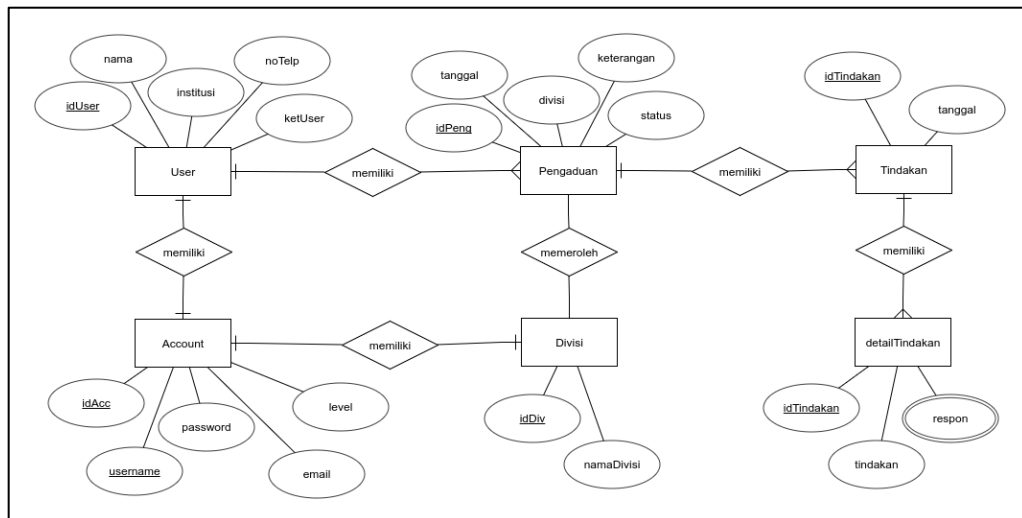
Gambar 12. Activity Diagram Untuk Kelola Data Tindak Lanjut Pengaduan

Complainee dapat melihat daftar pengaduan yang masuk secara keseluruhan maupun berdasarkan statusnya dengan kegiatan aplikasi dalam menanganinya terdapat pada Gambar 13.



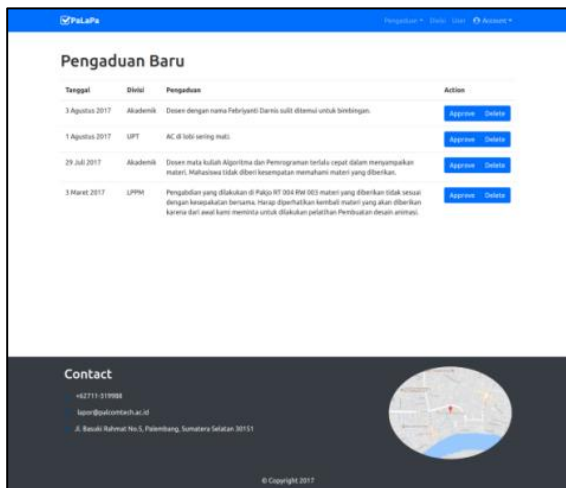
Gambar 13. Activity Diagram untuk Melihat Pengaduan yang Masuk (berdasarkan Status)

Gambar 14 memperlihatkan perancangan database yang digambarkan menggunakan Entity Relationship Diagram.



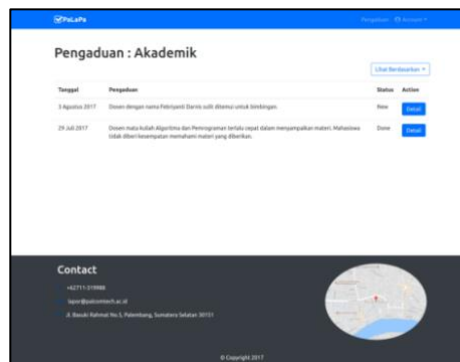
Gambar 14. Entity Relationship Diagram untuk Aplikasi PaLaPa

Beberapa hasil tampilan aplikasi diperlihatkan pada gambar dibawah ini. Gambar 15 memperlihatkan tampilan aplikasi menu untuk admin dalam melihat daftar pengaduan baru. Admin dapat menyetujui maupun menghapus pengaduan.



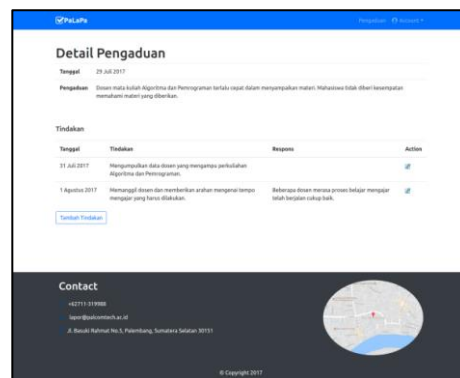
Gambar 15. Tampilan Aplikasi untuk Menu Pengaduan Baru untuk Admin

Ketika admin menyetujui sebuah pengaduan, pengaduan akan dapat diakses pada daftar pengaduan sesuai dengan divisinya. Gambar 16 memperlihatkan daftar seluruh pengaduan yang diterima oleh divisi akademik. Operator divisi dapat melihat daftar pengaduan berdasarkan statusnya.



Gambar 16. Tampilan Aplikasi untuk Menampilkan Daftar Pengaduan yang Masuk pada Divisi Akademik

Contoh detail pengaduan diperlihatkan pada Gambar 17.



Gambar 17. Tampilan Aplikasi untuk Detail Pengaduan

Pengujian dilakukan menggunakan metode *black box* dengan memasukkan data normal sebagai masukannya. Tabel 2 memperlihatkan beberapa hasil pengujian pengolahan data pada aplikasi.

Tabel 2. Hasil Pengujian Menggunakan Metode *Black Box*

Data Masukan	Luaran/Proses Yang Diharapkan	Pengamatan	Kesimpulan
Form buat pengaduan.	Ketika tombol simpan diklik, sistem akan menyimpan data pengaduan ke dalam database. Tanggal secara otomatis akan mengambil dari sistem. Jika terdapat satu field yang tidak diisi pada form, maka akan muncul notifikasi kesalahan.	Notifikasi pengaduan tersimpan muncul ketika seluruh field diisi. Notifikasi kesalahan muncul ketika terdapat satu field yang kosong.	Diterima.
Form lihat pengaduan per-divisi.	Jika pengguna login sebagai kepala divisi, maka pengaduan yang akan ditampilkan adalah pengaduan yang	Ketika login sebagai kepala divisi akademik, pengaduan yang ditampilkan adalah seluruh pengaduan yang	Diterima.

		ditujukan untuk divisi tersebut.	ditujukan mengenai kegiatan akademik.
Form tindakan.	memasukkan	Di form detail pengaduan, kepala divisi dapat memasukkan tindakan yang dilakukan mengenai sebuah pengaduan. Di form detail pengaduan pada akun pembuat pengaduan, dapat dilihat detail tindakan yang telah dimasukan oleh kepala divisi.	Ketika login sebagai kepala divisi akademik, tindakan yang dimasukkan tersimpan dalam database. Ketika login sebagai pembuat pengaduan, detail pengaduan menampilkan tindakan yang telah tersimpan di database.
			Diterima.

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa:

Extreme Programming dapat diterapkan untuk mengembangkan aplikasi yang digunakan untuk menampung keluhan pelanggan terhadap Perguruan Tinggi.

1. Penerapan *Extreme Programming* pada pengembangan aplikasi mengakibatkan pengembangan dapat dilakukan secara cepat dengan jumlah anggota tim yang minimal.
2. Aplikasi PaLaPa dapat digunakan untuk menampung keluhan pelanggan Perguruan Tinggi tidak hanya bersumber kepada mahasiswa saja, melainkan juga dapat berasal dari orang tua/wali, pihak pengguna lulusan, dan pihak masyarakat secara umum yang berinteraksi dengan PT.
3. Tahapan pengembangan aplikasi yang dilakukan hanya sebatas tahapan pengujian aplikasi menggunakan metode *blackbox*.
4. Dari hasil pengujian aplikasi dapat dijalankan sebagaimana mestinya dan memenuhi *acceptance criteria* pada *user story*.

5. SARAN

Beberapa saran yang dapat dijadikan acuan untuk pengembangan aplikasi selanjutnya adalah:

1. Perlunya dilakukan penyesuaian aplikasi terhadap permasalahan yang sebenarnya ditemui di Perguruan Tinggi.
2. Perlunya dilakukan usability testing untuk menguji apakah aplikasi telah sesuai dengan kebutuhan pengguna atau belum.

DAFTAR PUSTAKA

ADELIN dan EFFENDI, H., 2017. Aplikasi Audit Mutu Akademik Internal dengan Pendekatan Extreme Programming. *Jurnal TI Atma Luhur*, [e-journal] 4(1), p.13-24. Tersedia di: <<http://jurnal.atmaluhur.ac.id/index.php/TI>

<[_atma_luhur/article/view/264/215](http://atma_luhur/article/view/264/215)>

[Diakses pada 4 Agustus 2017]

AITKEN A. dan ILANGO, V., 2013. A Comparative Analysis of Traditional Software Engineering and Agile Software Development. *Hawaii International Conference on System Sciences*, [Online]. Tersedia di: <<https://pdfs.semanticscholar.org/6fb5/bf372782860a799cd8e9a94e77b371a9c2d2.pdf>> [Diakses pada 4 Agustus 2017]

BAKRI, T., 2016. Aplikasi Pengaduan Pelayanan Prima Resmi Diluncurkan. Website Universitas Tadulako, [Online]. Tersedia di: <<http://untad.ac.id/aplikasi-pengaduan-pelayanan-prima-resmi-diluncurkan/>> [Diakses pada 27 Juli 2017]

CHOUDHARI, J. dan SUMAN, U., 2012. Designing RC Story for Software Maintenance and Evolution. *Journal of Software*, [e-journal] 7(5), p.1103-1108. Tersedia di: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.850.3722&rep=rep1&type=pdf>> [Diakses pada 10 Oktober 2017]

DULLEMOND, K. dan GAMEREN, B.v., 2009. Technological Support for Distributed Agile Development, [M.S. Thesis]. Tersedia di: <<https://repository.tudelft.nl/islandora/object/uuid:cb213f3a-f6f5-41e0-b56f-07f13f37af22/datastream/OBJ>> [Diakses pada 2 September 2017]

FERDIANSYAH, M.S., JASRI, M., dan WIDJIANTO, 2016. Aplikasi Quick Response dalam Melayani Pengaduan Kerusakan Sarana STT Nurul Jadid Berbasis Android dan Web. *Prosiding SENTIA 2016*, [e-Proceeding] p.A-152-A-157. Tersedia di: <<http://sentia.polinema.ac.id/index.php/SENTIA2016/article/viewFile/32/27>> [Diakses pada 29 Juli 2017]

GUNAWAN, H. dan AGUSTIAN, I., 2014. Aplikasi Pelayanan Gangguan Listrik Berbasis Web

di PLN Rayon Banjaran. *Jurnal Informasi*, [e-journal] 6(2), p.1-15. Tersedia di: <<http://informasi.stmik-im.ac.id/wp-content/uploads/2016/05/hendra62-1.pdf>> [Diakses pada 29 Juli 2017]

HANNAY, J.E., ARISHOLM, E., ENGVIK, H. dan SJOBERG, D.I., 2010. Effects of Personality on Pair Programming. [Online]. Tersedia di: <<https://pdfs.semanticscholar.org/c59c/0740c5258b47ac95466c4ca4a4b9627699c1.pdf>> [Diakses pada 12 November 2017]

MASYA, F., ELVINA dan SIMANJUNTAK, F.M., 2012. Sistem Pelayanan Pengaduan Masyarakat pada Divisi Humas Polri Berbasis Web. *Prosiding Seminar Nasional Aplikasi Teknologi Informasi 2012*, Yogyakarta, [e-Proceeding]. Tersedia di: <<http://journal.uui.ac.id/index.php/Snati/article/view/2897/2656>> [Diakses pada 29 Juli 2017]

PRABOWO, S.A., SHOLIQ dan MUQTADIROH, F.A., 2013. Rancang Bangun Aplikasi Web Inforasi Eksekutif pada Pemerintahan Kabupaten XYZ. *Jurnal Teknik POMITS*, [e-journal] 2(3), p.A-476-A480. Tersedia di: <<http://ejurnal.its.ac.id/index.php/teknik/article/view/5192/1544>> [Diakses pada 4 Agustus 2017]

RAMADHANI, A.N., ADAM P., Y. dan ADI. T.N., 2015. Pembangunan Aplikasi Web Distribusi Kelompok Tani Katata dengan Menggunakan Metode Extreme Programming. *e-Proceeding of Engineering*, [e-proceeding] 2(2), p.5445-5452. Tersedia di: <https://openlibrary.telkomuniversity.ac.id/pustaka/files/101996/jurnal_eproc/pembangunan-aplikasi-web-distribusi-kelompok-tani-katata-dengan-menggunakan-metode-extreme-programming.pdf> [Diakses pada 27 Juli 2017]