

Pengembangan e-disposisi Menggunakan Metode Extreme Programming (Studi Kasus BPKAD Kotawaringin Timur)

Slamet Riyadi¹

¹Sistem Informasi, Fakultas Ilmu Komputer Universitas Darwan Ali, Sampit
E-mail: *¹slamet.riau2@gmail.com

Abstrak – Proses disposisi surat yang ada di BPKAD Kotawaringin Timur selama ini masih mengandalkan proses manual. Proses manual tersebut tentu tidak akan luput dari human error maupun sulitnya melacak history disposisi surat serta dapat menyulitkan proses pengarsipan. Dari permasalahan tersebut tentu sangat dibutuhkan sebuah aplikasi sistem informasi yang dapat digunakan untuk menangani proses disposisi surat. Pilihan yang sangat mungkin diterapkan yaitu mengembangkan sistem dengan penunjukan langsung/melakukan kerjasama dengan sedikit tenaga programmer sesuai anggaran yang dimiliki.

Untuk mengembangkan system informasi tersebut perlu ditentukan metode pengembangan yang sesuai dengan kondisi instansi. Dari beberapa metode yang ada, Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan-perubahan requirement yang sangat cepat.

Metode extreme programming dapat digunakan untuk membuat aplikasi e-disposisi dengan segala keterbatasan sumber daya baik dari sisi SDM maupun anggaran. Hasil pengujian menunjukkan bahwa aplikasi ini sudah memenuhi acceptance criteria yang tertuang pada user stories, sehingga aplikasi ini layak untuk digunakan.

Kata Kunci — Extreme Programming, BPKAD, e-disposisi

Abstract – The process of disposition of letters in BPKAD Kotawaringin Timur so far still relies on a manual process. The manual process will certainly not escape human error or the difficulty of tracking the disposition of letter history and can complicate the archiving process. Of these problems, of course an information system application is needed that can be used to handle the letter disposition process. The choice that is very likely to be implemented is to develop the system by direct appointment with a small number of programmers according to their budget.

To develop the information system, it is necessary to determine a development method that is appropriate to the conditions of the agency. Of the several methods available, Extreme Programming is a software engineering process that tends to use an object oriented approach and the target of this method is a team formed on a small to medium scale.

The extreme programming method can be used to create e-disposition applications with all the limitations of resources both in terms of HR and budget. The test results show that this application meets the acceptance criteria set forth in the user stories, so this application is fit for use.

Keywords — Extreme Programming, BPKAD, e-disposisi

1. PENDAHULUAN

Surat merupakan alat komunikasi tertulis yang berasal dari satu pihak dan ditujukan kepada pihak lain untuk menyampaikan warta [1]. Sedangkan menurut Gie [2], surat adalah setiap bentuk catatan tertulis atau bergambar yang memuat keterangan mengenai sesuatu hal atau peristiwa yang dibuat orang untuk membantu ingatannya. Surat merupakan kertas dan sebagainya yang bertulis atau secarik kertas dan sebagainya sebagai tanda atau sesuatu yang ditulis[3].

Disposisi menjadi kegiatan opsional setelah pendistribusian surat, dimana langkah-langkah pendistribusian surat adalah menyampaikan surat-surat ke alamat yang dituju, mengklasifikasikan surat dengan menyusun berdasarkan tingkat kepentingannya dan menentukan pejabat mana yang harus menangani surat dengan menyertakan lembar disposisi[4]. Disposisi merupakan petunjuk singkat tentang tindak lanjut (penyelesaian) terhadap suatu urusan atau surat masuk[5].

Proses disposisi surat yang ada di BPKAD (Badan Pengelola Keuangan dan Aset Daerah) Kotawaringin Timur selama ini masih mengandalkan proses manual. Dimana proses tersebut dimulai dari admin SOPD (Satuan Operasional Perangkat Daerah) datang ke BPKD membawa surat dan menyampaikannya kepada admin Kaban (Kepala Badan). Admin akan menverifikasi surat untuk selanjutnya disampaikan kepada Kaban untuk diberikan disposisi kepada bawahannya. Disposisi akan diteruskan kepada pihak-pihak terkait hingga proses tersebut selesai.

Proses manual tersebut tentu tidak akan luput dari human error maupun sulitnya melacak history disposisi surat serta dapat menyulitkan proses pengarsipan. Dari permasalahan tersebut tentu sangat dibutuhkan sebuah aplikasi sistem informasi yang dapat digunakan untuk menangani proses disposisi surat.

2. METODE PENELITIAN

Keterbatasan sumber daya yang dimiliki instansi baik dari segi anggaran maupun SDM membuat instansi harus memilih alternatif untuk dapat mengembangkan sistem informasinya. Dari beberapa pilihan yang ada, membeli product/software yang sudah jadi maupun kerjasama dengan perusahaan/pihak ketiga memerlukan biaya yang sangat tinggi. Oleh sebab itu, pilihan yang mungkin diterapkan yaitu mengembangkan sistem dengan penunjukan langsung/melakukan kerjasama dengan sedikit tenaga programmer sesuai anggaran yang dimiliki.

Untuk mengembangkan sistem informasi tersebut perlu ditentukan metode pengembangan yang sesuai dengan kondisi instansi. Dari beberapa metode yang ada, Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan-perubahan requirement yang sangat cepat[6].

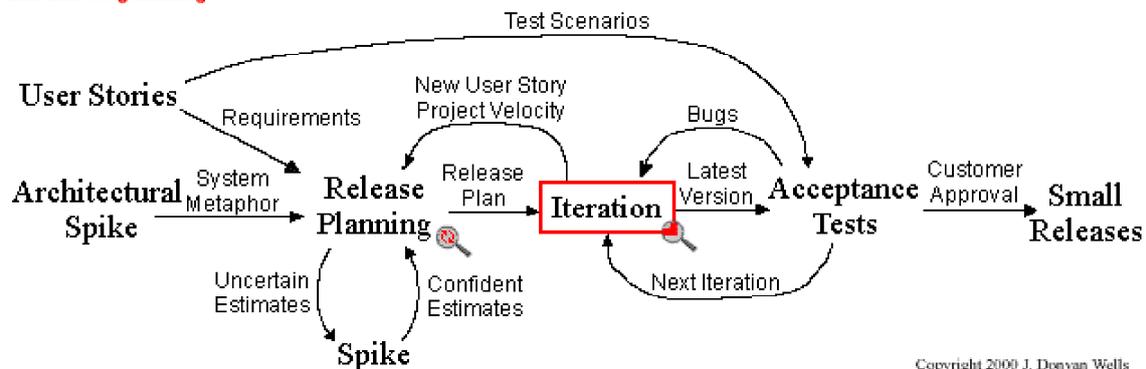
Metode XP pertama kali ditemukan oleh Kent Beck, seorang pakar software engineering yang bekerja dengan Chrysler yang membuat proyek C3 (Chrysler Comprehensive Compensation). Pada saat itu proyek Chrysler terancam gagal sebelum Kent Beck dikontrak, kemudian saat Kent Beck memegang proyek ini, Kent Beck bersama Ron Jeffries menyelesaikan proyek sesuai dengan target dengan menggunakan berbagai metode, kemudian kumpulan metode ini disebut dengan metode XP. Kent Beck membuat perubahan pada pembangunan proyek dengan membuatnya lebih efisien, adaptif, dan fleksibel[7].

Penelitian yang dilakukan oleh Ariaji[8], pemilihan metode extreme programming dalam pengembangan sistem informasi peminjaman alat adalah benar berdasarkan evaluasi menggunakan radar chart Boehm Turner. Tingkat keberhasilan sistem informasi peminjaman alat laboratorium jika diukur menggunakan kepuasan pengguna akhir (EUCS) pada level puas sehingga dapat disimpulkan sistem informasi peminjaman alat adalah berhasil.

Aturan pengembangan sistem informasi menggunakan metode Extreme Programming terdiri dari beberapa tahapan yaitu *Planning*, *Managing*, *Designing*, *Coding*, serta *Testing* sebagaimana terdapat pada gambar 1[9].



Extreme Programming Project



Copyright 2009 J. Donvan Wells

Gambar 1. Siklus Pengembangan dalam metode Extreme Programming

Adapun tahapan detail dari Extreme Programming yaitu[10]:

1. Planning (perencanaan), merupakan tahap pertama yang dilakukan oleh sebuah tim dalam mengembangkan sebuah sistem. Aktivitas planning pada model proses XP berfokus pada mendapatkan gambaran fitur serta fungsi dari perangkat lunak yang akan dibangun. Pada aktivitas ini dimulai dengan membuat kumpulan cerita atau gambaran yang diberikan klien (user stories) yang kemudian akan menjadi gambaran dasar dari perangkat lunak.
2. Design (perancangan), merupakan tahap perancangan system yang akan dibuat sesuai dengan user story pada tahap pertama. Aktivitas design dalam pengembangan aplikasi bertujuan untuk mengatur pola logika dalam sistem. Sebuah design yang baik, dapat mengurangi ketergantungan antar setiap proses pada sebuah sistem. Dengan begitu, jika salah satu fitur pada sistem mengalami kerusakan, tidak akan mempengaruhi sistem secara keseluruhan. Design pada model proses extreme programming menjadi panduan dalam membangun perangkat lunak yang didasari dari cerita klien sebelumnya. Dalam XP, proses design terjadi sebelum dan sesudah aktivitas coding berlangsung. Yang berarti aktivitas design terjadi secara terus-menerus selama proses pengembangan aplikasi berlangsung.
3. Coding (pengkodean), merupakan tahap implementasi dari design yang telah dibuat. Setelah menyelesaikan pengumpulan cerita dan menyelesaikan design untuk aplikasi secara keseluruhan, XP lebih merekomendasikan tim untuk terlebih dahulu membuat modul unit tes yang bertujuan untuk melakukan uji coba setiap cerita yang didapat dari klien. Setelah berbagai unit tes selesai dibangun, tim barulah melanjutkan aktivitasnya ke penulisan coding aplikasi. XP menerapkan konsep pair programming dimana setiap tugas sebuah modul dikembangkan oleh dua orang programmer. XP beranggapan, 2 orang akan lebih cepat dan baik dalam menyelesaikan sebuah masalah. Selanjutnya, modul aplikasi yang sudah selesai dibangun akan digabungkan dengan aplikasi utama.
4. Testing (pengujian), merupakan tahap pengujian sistem. Tahapan uji coba pada XP sudah dilakukan juga pada saat tahapan sebelumnya yaitu coding. XP menerapkan perbaikan masalah kecil dengan sesegera mungkin akan lebih baik dibandingkan menyelesaikan masalah pada saat akan mencapai tenggat akhir. Oleh karena itu, setiap modul yang sedang dikembangkan akan terlebih dahulu mengalami pengujian dengan modul unit tes yang telah dibuat sebelumnya. Setelah semua modul telah dikumpulkan dalam sebuah sistem yang sempurna, barulah pengujian penerimaan (*acceptance test*) dilakukan. Pada tahapan pengujian ini aplikasi langsung diuji coba oleh pengguna atau klien dan mendapat tanggapan langsung mengenai penerapan cerita yang telah digambarkan sebelumnya. Hasil dan Pembahasan

Pengembangan aplikasi yang didasarkan pada metode extreme programming dimulai dengan tahapan planning. Pada tahap ini pengguna menguraikan kebutuhan-kebutuhannya yang nantinya akan dituangkan dalam bentuk user story. User story menggambarkan fungsionalitas sistem atau software yang sangat penting bagi pengguna [11].

Untuk menulis user stories yang baik, Bill Wake [12] merancang akronim INVEST sebagai pengingat dalam pembuatan user stories. INVEST merupakan akronim dari Independent, Negotiable, Valuable, Estimable, Small, dan Testable.

Independent berarti user stories tidak boleh tergantung pada user stories lainnya. Negotiable berarti stories harus menangkap esensi dari kebutuhan dan tidak harus mewakili kontrak, jadi apa yang ada di dokumen kontrak/requirement tidak harus diimplementasikan ke dalam software. Valuable berarti stories harus memiliki nilai yang jelas bagi pelanggan, bukan bagi developer. Estimable berarti dapat diperkirakan sehingga dapat dibuat jadwal implementasi stories tersebut. Small berarti stories secara umum merepresentasikan beberapa/sedikit orang dalam beberapa minggu kerja. Testable, cara yang efektif untuk memastikannya adalah dengan menentukan kriteria user acceptance untuk semua stories.

User stories dapat ditulis dengan terdiri dari beberapa bagian yaitu judul, deskripsi, serta Acceptance Criteria[13]. Rangkuman user stories pada penelitian ini sebagaimana tertuang dalam tabel 1 didasarkan pada kebutuhan user.

Tabel 1. User Stories e-disposisi

No.	Judul	Deskripsi	Acceptance Criteria
1	Kirim surat	Sebagai user SOPD, saya ingin dapat mengirimkan surat	Adanya menu yang dapat digunakan untuk mengupload surat.
2	Penerima surat baru	Sebagai admin, saya ingin dapat memilih pejabat yang pertama kali menerima surat baru ketika kepala badan sedang berhalangan	Adanya menu untuk mengatur penerima surat baru
3	Approve surat	Sebagai admin, saya ingin dapat memverifikasi setiap surat baru kemudian meneruskannya kepada pejabat yang berwenang menerima surat baru	Adanya menu approve surat baru yang masuk sehingga dapat diverifikasi validitas surat tersebut
4	Daftar surat/disposisi masuk	Sebagai pejabat, saya ingin dapat melihat surat baru maupun disposisi baru yang saya terima	Adanya menu untuk melihat daftar surat baru maupun disposisi baru yang ditindak lanjuti
5	Buat Disposisi	Sebagai pejabat, saya ingin dapat menuliskan disposisi dan meneruskannya kepada pejabat lainnya	Adanya form/menu untuk menuliskan disposisi dan meneruskannya kepada pejabat lainnya
6	History Disposisi	Sebagai admin, saya ingin dapat melihat riwayat disposisi suatu surat	Adanya menu untuk menampilkan riwayat disposisi setiap surat
7	Balas Surat	Sebagai admin, saya ingin dapat mengirimkan surat balasan kepada user SOPD	Adanya menu untuk mengirimkan surat balasan
8	Surat Balasan	Sebagai user SOPD, saya ingin dapat melihat surat balasan dari admin BPKAD	Adanya menu untuk melihat surat balasan

Setelah user stories disusun, tahap selanjutnya yaitu proses perancangan/design yang berdasarkan stories tersebut. Untuk pemodelan design aplikasi ini digunakan diagram yang ada pada

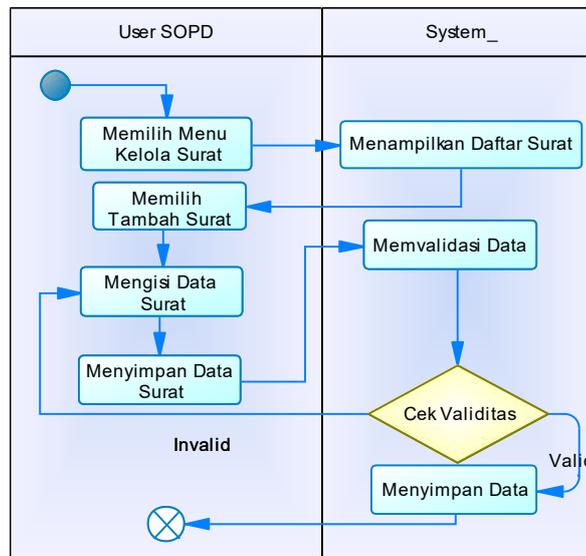
UML (Unified Modeling Language). Gambar 1 menunjukkan use case diagram e-disposisi yang merupakan pemodelan dasar dari acceptance criteria yang ada pada user stories di



Gambar 1. Use Case diagram e-disposisi

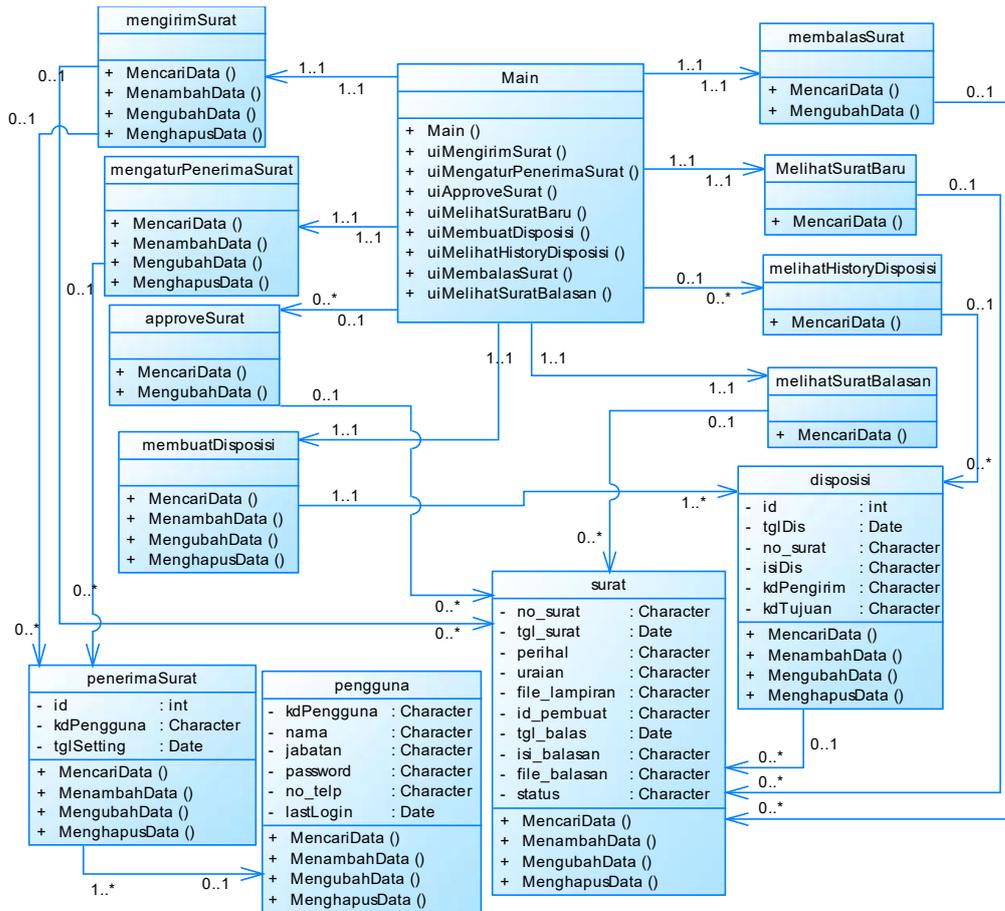
Dari use case diagram diatas, selanjutnya setiap case akan digambarkan secara detail menggunakan activity diagram. Penggambaran activity tersebut mengacu kepada user story yang telah disusun sebelumnya.

Salah satu activity kirim surat terdapat pada gambar 2. Gambar tersebut menjelaskan detail langkah-langkah proses pengiriman surat yang dilakukan oleh user SOPD. Langkah ini dimulai ketika user memilih menu kelola surat dan dilanjutkan pengisian surat serta menyimpannya ke dalam database.



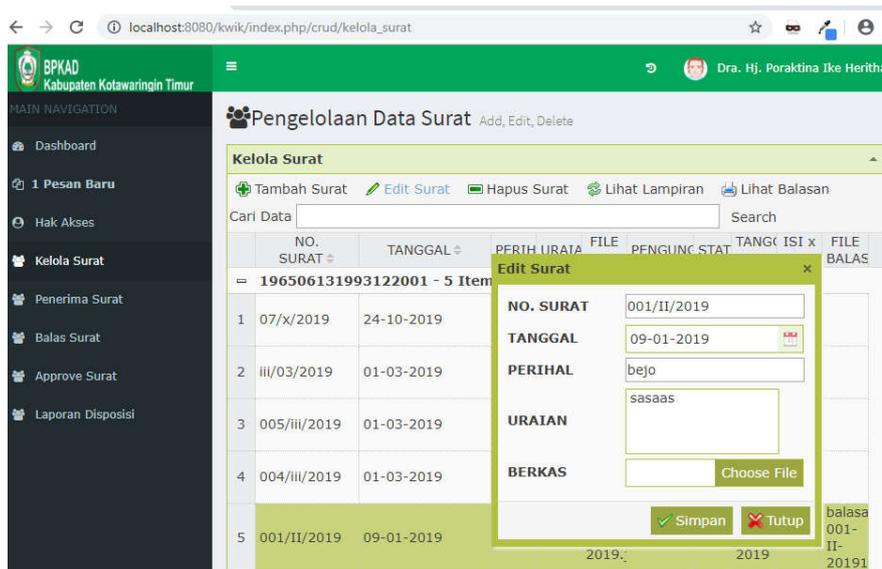
Gambar 2. Activity Diagram mengirim surat

Setelah semua detail proses yang ada di aplikasi digambarkan, langkah selanjutnya yaitu proses pembuatan class diagram yang merupakan indentifikasi semua class yang ada pada semua activity sebelumnya. Class diagram ini dapat langsung menggambarkan desain database yang dibutuhkan aplikasi. Adapun detail dari class diagram ini seperti pada gambar 3.



Gambar 3. Class Diagram e-disposisi

Setelah semua tahapan desain selesai, tahap selanjutnya yaitu proses coding. Tahapan ini merupakan implementasi dari desain sebelumnya. Adapun salah satu form hasil tahap coding seperti pada gambar 4 dibawah ini



Gambar 4. Form Pengelolaan Data Surat

Setelah tahapan coding selesai tahap selanjutnya yaitu pengujian. Adapun pengujian yang dilakukan yaitu menggunakan metode blackbox.

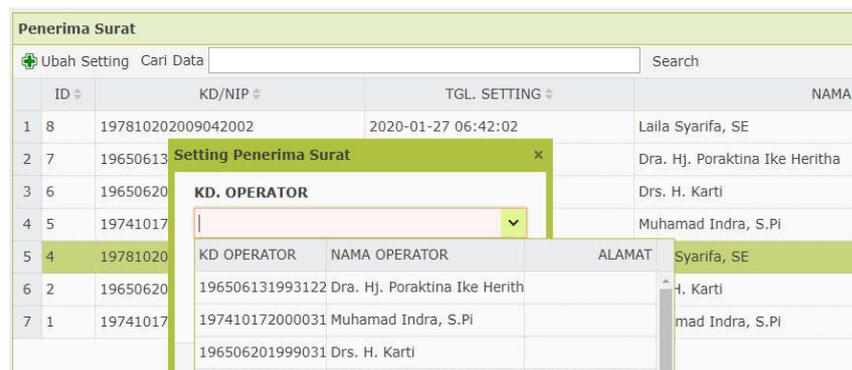
Black-Box Testing juga disebut sebagai pengujian fungsional, pengujian fungsional teknik yang merancang uji kasus berdasarkan informasi dari spesifikasi fungsional program. Dengan black box testing, tester seharusnya tidak memiliki akses ke kode sumber internal itu sendiri. Pengujian black box berfokus pada output yang dihasilkan sebagai respons terhadap input yang dipilih dan kondisi yang eksekusi [14]. Salah satu jenis pengujian yang ada di black box yaitu Boundary value analysis. Analisis ini dilakukan dengan membuat tes yang memuat bawah dan batas atas dari setiap inputan. Tes ini dapat digunakan untuk mengetahui secara detail keberhasilan dari setiap data yang diinput[15].

Tabel 2. Pengujian Form Kirim Surat

Field Uji	Data Sampel	Hasil yang Diharapkan	Hasil	Kesimpulan
No. Surat	(Dikosongkan)	False	True	Sukses
	(No.Baru) 001/X/2019	True	True	Sukses
	(No.Sama) 001/X/2019	False	True	Sukses
Tanggal	(Dikosongkan)	False	True	Sukses
	(Batas bawah) 31-12-2018	False	True	Sukses
	(Batas atas) 01-01-3000	True	True	Sukses
Perihal	(Dikosongkan)	False	True	Sukses
	(Batas bawah, 1 karakter)	True	True	Sukses
	(Batas atas, > 50 karakter)	False	True	Sukses
Uraian	(Dikosongkan)	False	True	Sukses
	(Batas bawah, 1 karakter)	True	True	Sukses
	(Batas atas, > 100 karakter)	False	True	Sukses
Berkas	(Dikosongkan)	False	True	Sukses
	(File Lampiran < 20 MB)	True	True	Sukses
	(File Lampiran > 20 MB)	False	True	Sukses

Dari hasil pengujian pada tabel 2 dapat dijelaskan bahwa setiap inputan wajib diisi. No. Surat merupakan field kunci sehingga tidak diizinkan adanya duplikat no. surat. Sedangkan untuk inputan lainnya memiliki batasan yang ada seperti pada tabel tersebut. Dari hasil penginputan semua field menunjukkan tidak adanya kegagalan sehingga dapat disimpulkan bahwa form ini layak digunakan untuk penginputan data.

Untuk pengujian selanjutnya yaitu Form Setting Penerima Surat. Gambar 5 menunjukkan bahwa pencarian data penerima surat baru dapat dilakukan yang selanjutnya dapat dipilih dan disimpan



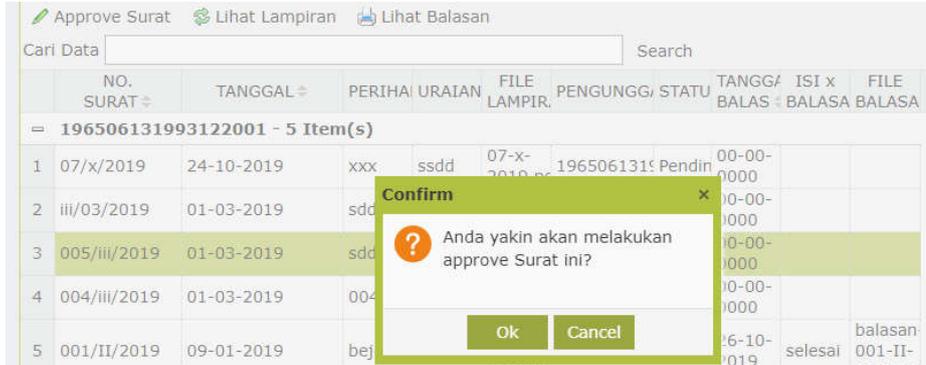
Gambar 5. Form Setting Penerima Surat

Tabel 3. Pengujian Form Setting Penerima Surat

Field Uji	Data Sampel	Hasil yang Diharapkan	Hasil	Kesimpulan
Kd. Operator	(Dikosongkan)	False	True	Sukses
	(Diketik pencarian)	True	True	Sukses
	(Dipilih penerima)	True	True	Sukses
Tombol	(Tidak diklik)	False	True	Sukses

Simpan	(Diklik) Menyimpan data	True	True	Sukses
Tombol Batal	(Tidak diklik)	False	True	Sukses
	(Diklik) Menutup form	True	True	Sukses

Dari hasil pengujian pada tabel 3 menunjukkan bahwa form setting penerima surat sudah berfungsi sebagaimana yang dibutuhkan user. Untuk pengujian selanjutnya yaitu menu approve surat. Adapun form approve surat yang akan diuji seperti pada gambar 6.

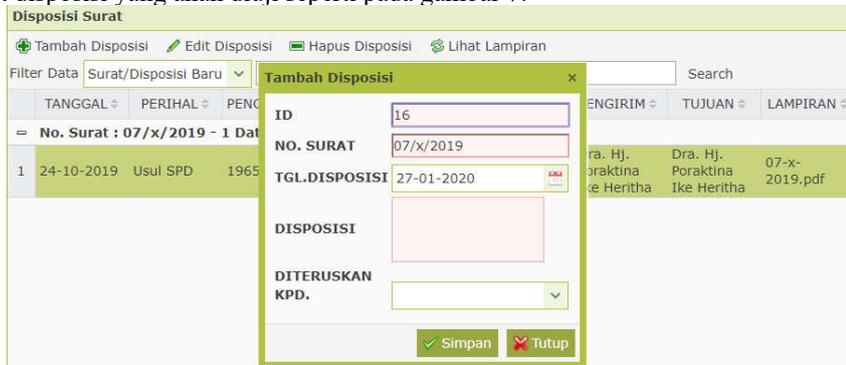


Gambar 6. Form Approve Surat

Tabel 4. Pengujian Form Approve Surat

Field Uji	Data Sampel	Hasil yang Diharapkan	Hasil	Kesimpulan
Tombol Approve Surat	(Tidak diklik)	False	True	Sukses
	(Diklik) Muncul Konfirmasi	True	True	Sukses
Tombol Ok	(Tidak diklik)	False	True	Sukses
	(Diklik) Menyimpan data	True	True	Sukses
Tombol Cancel	(Tidak diklik)	False	True	Sukses
	(Diklik) Menutup form	True	True	Sukses

Dari hasil pengujian pada tabel 4 menunjukkan bahwa form approve surat sudah berfungsi sebagaimana yang dibutuhkan user. Untuk pengujian selanjutnya yaitu menu buat disposisi. Adapun form buat disposisi yang akan diuji seperti pada gambar 7.

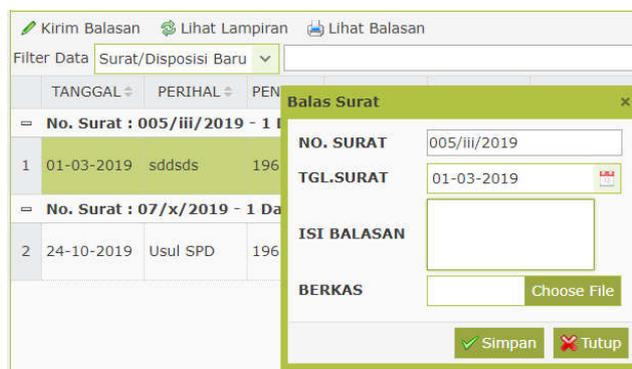


Gambar 7. Form Buat Disposisi

Tabel 5. Pengujian Form Buat Disposisi

Field Uji	Data Sampel	Hasil yang Diharapkan	Hasil	Kesimpulan
Tanggal	(Dikosongkan)	False	True	Sukses
	(Batas bawah) 31-12-2018	False	True	Sukses
	(Batas atas) 01-01-3000	True	True	Sukses
Disposisi	(Dikosongkan)	False	True	Sukses
	(Batas bawah, 1 karakter)	True	True	Sukses
	(Batas atas, > 100 karakter)	False	True	Sukses
Diteruskan Kepada	(Dikosongkan)	False	True	Sukses
	(Pencarian data ditemukan)	True	True	Sukses
	(Pencarian data tidak ditemukan)	False	True	Sukses

Dari hasil pengujian pada tabel 5 dapat dijelaskan bahwa setiap inputan wajib diisi. Untuk ID dan No. Surat bersifat read only, sehingga tidak dapat diubah. Untuk field diteruskan kepada dapat dilakukan pencarian datanya, apabila tidak ditemukan maka kata kuncinya dianggap false dan tidak dapat disimpan. Untuk pengujian selanjutnya yaitu form balas surat seperti pada gambar 8.



Gambar 8. Form Balas Surat

Tabel 6. Pengujian Form Balas Surat

Field Uji	Data Sampel	Hasil yang Diharapkan	Hasil	Kesimpulan
Isi Balasan	(Dikosongkan)	False	True	Sukses
	(Batas bawah, 1 karakter)	True	True	Sukses
	(Batas atas, > 100 karakter)	False	True	Sukses
Berkas	(Dikosongkan)	False	True	Sukses
	(File Lampiran < 20 MB)	True	True	Sukses
	(File Lampiran > 20 MB)	False	True	Sukses

Dari hasil pengujian pada tabel 6 dapat dijelaskan bahwa setiap inputan wajib diisi. Untuk No. Surat dan Tgl. Surat bersifat read only, sehingga tidak dapat diubah. Untuk berkas lampiran maksimal ukuran file sampai dengan 20 MB.

Dari seluruh pengujian yang sudah dilakukan diatas dapat disimpulkan bahwa semua form input data sudah berfungsi sebagaimana yang dibutuhkan oleh user. Kesimpulan dari tahap pengujian dalam extreme ini dapat dituangkan kedalam tabel 7 berikut.

Tabel 2. Rangkuman hasil pengujian aplikasi e-disposisi

Nama Menu	Output yang diharapkan	Hasil Pengamatan	Kesimpulan Akhir
Kelola Surat	dapat memenuhi Acceptance Criteria	Sudah sesuai dengan desain	Diterima
Setting Penerima Surat	dapat memenuhi Acceptance Criteria	Sudah sesuai dengan desain	Diterima

Kelola Disposisi	dapat memenuhi Acceptance Criteria	Sudah sesuai dengan desain	Diterima
Balas Surat	dapat memenuhi Acceptance Criteria	Sudah sesuai dengan desain	Diterima

3. SIMPULAN

Dari hasil penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Metode extreme programming dapat digunakan untuk membuat aplikasi e-disposisi dengan segala keterbatasan sumber daya baik dari sisi SDM maupun anggaran.
2. Tahapan pembuatan aplikasi ini sudah sesuai dengan tahapan yang ada pada metode extreme programming
3. Hasil pengujian menunjukkan bahwa aplikasi ini sudah memenuhi acceptance criteria yang tertuang pada user stories, sehingga aplikasi ini layak untuk digunakan.

4. SARAN

Beberapa saran untuk penelitian selanjutnya yaitu:

1. Perlunya dilakukan pengujian efektifitas dan efisiensi dari penerapan metode extreme programming untuk pembuatan e-disposisi terutama dalam hal jumlah SDM serta anggaran yang diperlukan.
2. Perlunya dilakukan perbandingan pengembangan aplikasi e-disposisi antara metode extreme programming dengan metode lain yang masih dalam satu lingkup agile method.
3. Untuk mendapatkan gambaran yang lebih mendalam terhadap user acceptance, dapat dilakukan penelitian menggunakan Technology Acceptance Model (TAM)

DAFTAR PUSTAKA

- [1]. Barthos, Basir. 2003. *Manajemen Kearsipan*, Bumi Aksara, Jakarta
- [2]. Gie, The Liang. 2000. *Administrasi Perkantoran*, Modern Liberty, Yogyakarta.
- [3]. <https://kbbi.kemdikbud.go.id/entri/surat>, diakses pada tanggal 01 Desember 2019.
- [4]. Herijanto, Pudji. 1994. *Modul Manajemen Kearsipan*. Malang, Politeknik Universitas Brawijaya, Malang.
- [5]. Vironica, Arie dan Sukadi, 2014. Rancang Bangun Aplikasi Pengelolaan Surat Masuk Dan Surat Keluar Pada Sekolah Menengah Pertama Negeri 2 Nawangan, *Speed Journal Volume 11 No 1*, 34-41.
- [6]. Beck, Kent. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 1999.
- [7]. Wells, D., *History of Extreme Programming*, www.extremeprogramming.org/donwells.html, diakses tanggal 01 Desember 2019.
- [8]. Ariaaji, T., Utami, E., Sunyoto, A., *Evaluasi Sistem Informasi Yang Dikembangkan dengan Metodologi Extreme Programming*, *Jurnal Ilmiah DASI Vol. 15 No. 04 Desember 2014*, hlm 53 – 62, ISSN: 1411-3201.
- [9]. Wells, D., *The Rules of Extreme Programming*, <http://www.extremeprogramming.org/rules.html>, diakses tanggal 01 Desember 2019.
- [10]. Pressman, R. S., 2010, *Software Engineering A Practitioner's Approach*, Seventh Edition, McGraw Hill Inc, New York.
- [11]. Cohn, Mike, 2009, *User Stories Appliedfor Agile Software Development*, Kent Beck, Indiana
- [12]. Wake, William, C., 2002, *Extreme Programming Explored*, Addison-Wesley Professional

- [13]. Choudhari, J., Suman, U., 2012. Designing RC Story for Software Maintenance and Evolution. *Journal of Software*, [e-journal] 7(5), p.1103-1108. Tersedia di: [Diakses pada 10 Oktober 2019]
- [14]. H. Liu and H. B. Kuan Tan, "Covering code behavior on input validation in functional testing," *Information and Software Technology*, vol. 51, no. 2, pp. 546–553, Feb. 2009.
- [15]. T. Murnane, K. Reed, and R. Hall, "On the Learnability of Two Representations of Equivalence Partitioning and Boundary Value Analysis," in *Software Engineering Conference, 2007. ASWEC 2007. 18th Australian, 2007*, pp. 274

