

Pengembangan *User Interface* Sistem Informasi *Planned Maintenance System* pada PT. Pertamina Trans Kontinental dengan Menggunakan Metode *Design Sprint*

Jelang R. Kharisma¹⁾ Puspita Kartikasari²⁾ Tri Sagirani³⁾

Program Studi/Jurusan Sistem Informasi
Institut Bisnis dan Informatika Stikom Surabaya
Jl. Raya Kedung Baruk 98 Surabaya, 60298

Email : 1) jelangrkharia@gmail.com, 2) tri.sagirani@gmail.com, 3) tri.sagirani@gmail.com

Abstract: PT. Pertamina Trans Kontinental (PTK) is company that active in a maritime logistics, therefore its vessels idle time during maintenance become a crucial aspects for PTK because it affect their core business activity directly. To support every maintenance agenda, PTK develop a computerized maintenance management system (*Planned Maintenance System*) that help maintaining every detail of the vessels. In its implementation, *User Interface Design* is an important aspect since UI bridge the interaction between human and the system built. A system implementation without considering interaction between user and the system will result in a difficult interaction between user and the system. By evaluating user interface's usability, It's expected that the UI recommendations will be developed in accordance with the problems faced by the users. The evaluation is held before and after the design process which use *Design Sprint Method*. The proposed design final evaluation conclude that there was a 24.13% increase in score, leveraging the average score from 2.50 to 3.29.

Keywords: *Design Sprint, Computerized Maintenance Management System, Usability, User Interface*

Berlokasi di Jalan Kramat Raya No. 29 Jakarta Pusat, PT Pertamina Trans Kontinental (PTK) didirikan pada tanggal 9 september 1969 sebagai anak perusahaan dari PT Pertamina. PTK Didirikan sebagai perusahaan yang bergerak dibidang industry jasa maritim untuk mendukung aktifitas PT Pertamina. Aktifitas Bisnis yang dijalankan pada PT Pertamina Trans Kontinental antara lain: Operasi Penyediaan Kapal, *Charter & Brokerage*, Keagenan Kapal, Pengelolaan Pelabuhan/*Port Management, Bunker Agent*, serta *Underwater Technics Hydro-Oceanography and Mapping*. Operasi penyediaan kapal merupakan *Core Business* dari PT Pertamina Trans Kontinental. Fasilitas armada kapal yang disediakan meliputi: *Oil Tanker, LPG Carrier, Anchor Handling and Tug Supply, Multi-Purpose Vessel, Harbor Tug, Tug Boat & Oil Barge, Straight Supply Vessel, Rigid Inflatable Boat, SPOB (Self Propelled Oil Barge)*.

Untuk mendukung *core bussiness* tersebut, PT PTK mengembangkan "*Planned Maintenance System(PMS)*". Suatu sistem informasi pemeliharaan kapal terkomputerisasi (*Computerized maintenance management system*) yang bertujuan untuk mengelola setiap

detil dari bagian kapal yang perlu dipelihara, membantu bagian pemeliharaan kapal dalam memutuskan prioritas perbaikan *part*, dan membantu ketelitian bagian pemeliharaan kapal. Desain *user interface* yang tidak dibuat dengan baik dapat mengakibatkan interaksi yang sulit antara pengguna dengan sistem, membuat pengguna frustrasi, dan performa pekerjaan yang buruk (Weichbroth & Sikorski, 2015).

Kullolli (2008) dalam penelitian yang bertajuk "*Selecting a Computerized Maintenance Management System*" mengutarakan bahwa salah satu diantara sekian fitur/indikator yang diutamakan dalam pemilihan CMMS adalah Kemudahan pemakaian. Jeff O'Brien (2008) mengutarakan bahwa salah satu fase dalam proses implementasi CMMS adalah melatih *End User* untuk menggunakan sistem sehingga Usabilitas CMMS yang diimplementasikan maksimal. Weichbroth & Sikorski (2015) mengutarakan bahwa celah usability merupakan suatu hal yang sulit diperbaiki, khususnya ketika baru terdeteksi pada fase akhir proyek. Oleh karena itu interaksi antara komputer dan manusia harus diperhatikan. *Computerized maintenance management system* yang dikembangkan oleh PT PTK saat ini masih

berada pada iterasi awal pengembangan sistem, dan melihat pentingnya kemudahan pemakaian *User Interface* dari sebuah sistem CMMS sehingga perlu dilakukan evaluasi dan perbaikan. Evaluasi dan perbaikan ini juga didorong oleh *user story* yang diperoleh penulis pada observasi dan wawancara pendahuluan yang dilakukan pada bulan Agustus 2018 bahwa: CMMS yang dibangun belum mencerminkan bahwa aplikasi tersebut terhubung langsung dengan Sistem Informasi pada PT PTK, keluhan mengenai pengaruh getaran kapal terhadap proses input data, serta belum adanya warning jika kapal berada pada kondisi *offline*. Beberapa temuan tersebut dirasa berdampak langsung terhadap kemudahan pemakaian sistem dan memotivasi penulis untuk menggali lebih dalam masalah yang timbul terkait usabilitas *user interface* sistem informasi *planned maintenance system* PT PTK.

Dengan merancang UI sistem informasi *Planned Maintenance System* pada PT. Pertamina Trans Kontinental menggunakan metode *Design Sprint*, diharapkan UI yang dibangun sesuai dengan masalah yang dihadapi oleh pengguna sistem informasi PMS sehingga

Maintenance

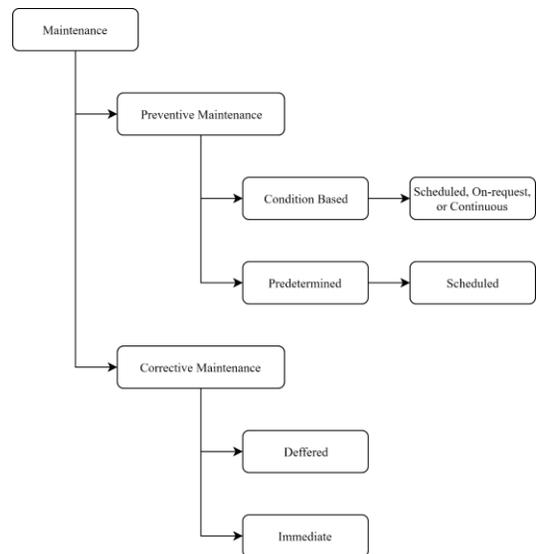
Berdasarkan terminologi Standar EN 13306:2010 (CEN, 2010) *Maintenance* didefinisikan sebagai kombinasi semua kegiatan teknis, administratif, dan manajerial selama *life-cycle* sebuah item yang bertujuan untuk mempertahankan, atau memperbaiki kondisi dimana item tersebut dapat berkerja sesuai fungsi yang diperlukan. Campbell (2015) dalam bukunya menemukan bahwa Biaya maintenance, (tergantung kondisi industri) dapat memakan setengah dari total biaya produksi. Selain itu, *down-time* yang disebabkan oleh proses *maintenance* yang buruk dapat menimbulkan biaya yang lebih besar.

Standar EN 13306:2010 (CEN, 2010) membagi *maintenance* menjadi *Preventive Maintenance* dan *Corrective Maintenance*. *Preventive Maintenance* sendiri terdiri dari dua jenis maintenance, yaitu *Condition Based Maintenance* dan *Predetermined Maintenance*. Sedangkan *Corrective Maintenance* dibagi berdasarkan kapan maintenance itu dilaksanakan, yaitu *maintenance* tertunda, atau *maintenance* langsung.

Preventive Maintenance dilakukan secara proaktif sebelum peralatan mengalami

kerusakan. *Maintenance* ini bertujuan untuk mengurangi peluang kerusakan dan degradasi fungsi dari sebuah peralatan. *Preventive Maintenance* merupakan *maintenance* yang terjadwal: *Predetermined Maintenance* dilaksanakan sesuai dengan program *maintenance* yang telah ditetapkan untuk setiap peralatan pada interval waktu tertentu. Sedangkan *Condition Based Maintenance* melihat kondisi peralatan yang dimonitor, kemudian menjadwalkan *maintenance* tergantung kondisi peralatan tersebut (CEN, 2010).

Corrective Maintenance dilakukan secara reaktif setelah kegagalan diidentifikasi, *maintenance* jenis ini bertujuan untuk mengembalikan kondisi peralatan menjadi kondisi yang dapat melakukan fungsi yang diperlukan (CEN, 2010). *Corrective Maintenance* biasanya dilakukan karena adanya kerusakan yang tidak dapat dibendung oleh *Preventive Maintenance*. Maintenance jenis ini cenderung memakan lebih banyak biaya, dan bisa jadi membutuhkan teknisi khusus dalam proses restorasi (Frenkel, Lisnianski, & Khvaskin, 2009).



Gambar 1. Gambaran umum *maintenance*

User Interface Prototyping

User interface prototyping merupakan suatu pendekatan yang dilakukan untuk melakukan pengujian dan evaluasi usabilitas pada konsep *User-Centered Design* (Weichbroth & Sikorski, 2015). Penulis juga sependapat dengan kutipan dari Dix et al (2004) dan Synder

(2003) oleh Weichbroth & Sikorski (2015) bahwa *User Interface prototyping* merupakan suatu pendekatan yang sangat baik untuk memfasilitasi komunikasi antara desainer dan pemangku kepentingan lain. *Prototyping* membantu dalam memberikan visualisasi konsep secara interaktif, sekaligus memberikan ekspektasi secara informal mengenai sistem yang akan dibuat.

Usability Evaluation

Usability didefinisikan oleh Nadikattu (2016) dalam tesisnya yang berjudul “*A Multiple Case Study Involving Organizations Developing Interactive Healthcare Technology Applications*” sebagai derajat kemudahan pengguna dalam berinteraksi dengan sistem, memenuhi kebutuhan pengguna, dan mempertahankan tingkat kepuasan pengguna yang tinggi, termasuk kemudahan pengguna tersebut dalam memahami dan mempelajari sistem. Dalam tesisnya, Nadikattu juga menyebutkan bahwa standarisasi mengenai usabilitas sistem terdokumentasi dalam ISO 9241. Dalam ISO 9241 usabilitas didefinisikan sebagai “sejauh mana sebuah produk dapat digunakan oleh suatu pengguna untuk mencapai sebuah tujuan secara efektif, efisien, dan puas dalam melakukan tugas yang spesifik. Komponen yang perlu digaris bawahi antara lain: Efektivitas, Efisiensi, dan Kepuasan.

Jeff Rubin dan Dana Chisnell (2008) mengutarakan bahwa *usability* mencerminkan kualitas dari sebuah produk. Beberapa aspek yang mendefinisikan usabilitas sebuah sistem antara lain *Efficiency*, *Effectiveness*, *Learnability*, dan *Satisfaction*. *Efficiency* merupakan seberapa cepat dan akurat seorang pengguna dapat menyelesaikan tugas dalam sebuah sistem. Efisiensi dihitung dalam unit waktu. *Effectiveness* mengacu pada sejauh mana sebuah sistem dapat merespon sesuai dengan ekspektasi pengguna. *Learnability* merupakan aspek semudah apa sebuah sistem dapat dikuasai oleh pengguna setelah dilakukan pelatihan. *Satisfaction* mengacu pada persepsi, perasaan, dan opini pengguna dalam menggunakan produk. Aspek *satisfaction* umumnya diukur melalui kuisisioner dan diskusi secara langsung.

Najmeh Ghasemifard et al (2015) mengutarakan bahwa *usability evaluation* merupakan metodologi yang digunakan untuk mengukur aspek-aspek usabilitas dari *user interface* sebuah sistem. Ivory & Hearst (2001)

mengutarakan bahwa *Usability Evaluation* adalah sebuah proses penting dalam pengembangan *User Interface*. Oleh karena itu *Usability Evaluation* diperlukan untuk memastikan tingkat kepuasan pengguna yang tinggi, mengurangi penolakan terhadap aplikasi yang dibuat, dan mengurangi peluang perubahan yang signifikan pada fase pengembangan aplikasi sehingga diharapkan dapat memangkas waktu, usaha, dan biaya pengembangan (Nadikattu, 2016). Pelaksanaan evaluasi *usability* sebaiknya dilakukan secara rutin (Silva, Silveira, Maurer, & Hellmann, 2012). Aktivitas utama dalam pelaksanaan *usability evaluation* antara lain: pengumpulan data usabilitas, analisis data, dan rekomendasi solusi (Ghasemifard, Shamsi, Rasouli Kenar, & Ahmadi, 2015). Dalam melaksanakan evaluasi usabilitas, beberapa hal yang harus diperhatikan antara lain (Conyer, 2008): (1) Tujuan dilaksanakannya evaluasi (2) Kriteria evaluasi. (3) Sejauh mana konstrain yang dikehendaki, apakah menggunakan isian, atau pilihan. (4) Metode Apa yang dipakai. (5) Jumlah, pengalaman, dan keahlian evaluator, dan pengguna. (6) Bagaimana hasil dari evaluasi dikomunikasikan. (7) Ketersediaan waktu dan biaya.

Conyer (2008) menyatakan ada beberapa metode yang dapat digunakan dalam melakukan evaluasi usabilitas, salah satunya adalah dengan menggunakan Kuisisioner. Evaluasi usabilitas menggunakan kuisisioner merupakan metode yang mudah dan terjangkau dalam pengumpulan dan pengolahan data. Kuisisioner yang menggunakan skala cenderung lebih mudah untuk dilakukan analisis kuantitatif, namun dapat mengurangi informasi yang dapat digali jika dibandingkan dengan kuisisioner isian bebas. Kuisisioner yang diberikan setelah menggunakan produk dapat meningkatkan peluang adanya informasi yang hilang jika dibandingkan ketika kuisisioner diberikan disaat yang sama ketika menggunakan produk/objek evaluasi.

Dalam proses pengumpulan *requirements* dapat dilakukan dengan melakukan *Focus Group Discussion* (FGD) yang berisi enam hingga sembilan orang. FGD dengan moderator dan narasumber yang tepat dapat menghasilkan ide, gagasan, dan opini terkait usabilitas, namun karena bentuk diskusi yang kualitatif, validitas data yang dikumpulkan menjadi suatu hal yang dipertanyakan.

(Ghasemifard, Shamsi, Rasouli Kenar, & Ahmadi, 2015)

Design Sprint

Design Sprint merupakan aktifitas/usaha untuk menyiapkan sebuah produk dengan mengedepankan tujuan dalam memvalidasi *prototype* dari produk yang akan dibuat. *Design Sprint* memperikan *roadmap* untuk memulai dan memvalidasi hampir semua pekerjaan mengenai produk digital yang dilaksanakan dalam 5 hari tiap iterasi (Banfield, Lombardo, & Wax, 2015).

Banfield et al (2015) mengutarakan bahwa *design sprint* dibagi dalam lima fase, yaitu *Understand*, *Diverge*, *Converge*, *Prototype*, dan *Test*. Fase *Understand* bertujuan untuk mengembangkan dan mendefinisikan wawasan terkait sebuah masalah. Fokus terhadap masalah apa yang sudah diidentifikasi, masalah apa yang sudah diidentifikasi, dan informasi apa saja yang diperlukan untuk mengatasi masalah tersebut. Fase *Diverge* merupakan fase lanjutan dari fase *Understand*, dimana dilakukan sesi untuk mengumpulkan berbagai macam jawaban dan solusi atas tantangan yang muncul dari masalah yang diidentifikasi sebelumnya. Fase *Diverge* dapat dilakukan dengan melakukan *Brainstorming* Ide untuk menyusun solusi. *Converge* merupakan fase dimana tim pengembang memeriksa setiap kemungkinan solusi untuk dipilih satu atau dua yang akan direalisasikan. Fase selanjutnya adalah fase membuat *Prototype* dengan memperhatikan asumsi dan temuan yang telah dibahas di fase sebelumnya. Fase yang terakhir adalah *Test*, dimana tim pengembang menguji apakah solusi yang diberikan benar-benar menyelesaikan masalah yang diidentifikasi sebelumnya. fase *Test* dapat dilakukan dengan melakukan *Testing Interview*.



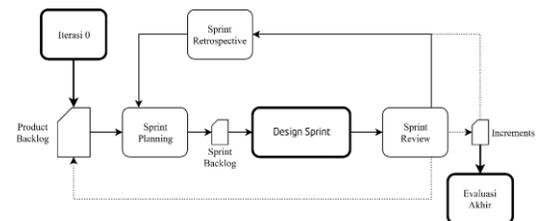
Gambar 2. Fase *Design Sprint*

METODE PENELITIAN

Penelitian ini menggunakan kerangka kerja *Scrum* untuk mengelola aktivitas yang dilaksanakan. Iterasi ke-0 dari penelitian ini dilakukan dengan melakukan survey evaluasi awal dan mengumpulkan *user story* terkait UI dari *Planned Maintenance System*. Evaluasi awal dilakukan dengan mengukur usabilitas dari UI yang sudah ada terhadap seluruh pengguna yang

terdaftar dalam sistem dengan memperhatikan Validitas dan Reliabilitas data. Pengukuran usabilitas diolah dengan menggunakan bantuan perangkat lunak SPSS. Kuisioner yang telah diolah kemudian dianalisa secara deskriptif.

Purwarupa yang dibuat pada fase *prototype* merupakan *High Fidelity Prototype* dengan menggunakan bantuan perangkat lunak *Adobe Illustrator*, *Photoshop*, dan *XD*.



Gambar 3. Metode Penelitian

Fase Pendahuluan (Iterasi 0)

Iterasi ke-0 dari penelitian ini dilakukan dengan melakukan identifikasi pengguna beserta aktivitas yang dilakukan didalam sistem, serta survey usabilitas awal sistem informasi PMS. Indikator usabilitas yang dievaluasi pada penelitian ini antara lain: (1) *Learnabilities*. (2) *Memmorabilities*. (3) *Efficiencies*. (4) *Errors*. (5) *Satisfactions*. Masing masing indikator usabilitas tersebut dibagi menjadi beberapa variabel untuk menyusun kuisioner . Responden diminta untuk menjawab setiap poin kuisioner dengan jawaban mulai dari Sangat Tidak Setuju=1, Tidak Setuju=2, Setuju=3, Sangat Setuju=4.

Variabel usabilitas yang dipakai dalam penelitian ini antara lain:

Tabel 1. Variabel & Indikator Usabilitas

Id	Uraian
<i>Learnabilities</i>	
P11	Saya dapat dengan mudah mengenali menu yang ada pada PMS
P12	Saya tidak membutuhkan waktu yang lama untuk mempelajari letak halaman pada PMS
P13	PMS Menggunakan bahasa yang sederhana dan mudah saya pahami
P14	Saya merasa bahwa Saya masih memerlukan pelatihan untuk mengoperasikan PMS
<i>Memmorabilities</i>	
P21	Saya dapat dengan mudah mengingat menu apa yang harus ditekan untuk menuju halaman tertentu

Id	Uraian
P22	Warna warna pada elemen halaman PMS mengingatkan saya pada <i>event</i> tertentu.
P23	Saya dapat mengingat letak form dan isi yang harus diisi pada form tersebut
P24	Tata letak Tampilan yang konsisten membantu saya mengingat elemen yang saya cari.
Efficiencies	
P31	Apa yang ditampilkan pada PMS membantu mempercepat pekerjaan saya.
P32	Saya dapat membaca teks yang ditampilkan pada halaman PMS dengan mudah
P33	Ukuran teks yang ditampilkan sudah baik
P34	Saya dapat dengan mudah berpindah dari halaman satu ke halaman yang lain
P35	Saya dapat mengisikan form ketika melakukan input dengan cepat
P36	Saya dapat dengan cepat menemukan informasi yang saya cari
P37	Saya dapat dengan mudah berinteraksi dengan tombol yang ditampilkan
P38	Saya dapat dengan mudah berinteraksi dengan link yang ditampilkan
P39	Saya dapat dengan mudah membedakan elemen mana yang dapat berinteraksi dengan saya
Errors	
P41	Ukuran teks yang ditampilkan memudahkan saya dalam mengenali kesalahan yang terjadi
P42	Pemilihan warna teks memudahkan saya dalam mengenali kesalahan yang terjadi
P43	Pemilihan warna pada tampilan pms memudahkan saya mengenali aktivitas diselesaikan dengan baik
P44	Pms menampilkan contoh hal yang benar ketika terjadi kesalahan
P45	Saya tidak membutuhkan bantuan orang lain ketika terjadi kesalahan
P46	PMS memberikan tampilan yang intuitif untuk meminimalisir kesalahan.
P47	Ketika saya melakukan kesalahan secara tidak sengaja, saya dapat memperbaiki sendiri kesalahan tersebut
Satisfactions	

Id	Uraian
P51	Secara garis besar, saya puas dengan informasi yang ditampilkan PMS
P52	Saya dapat membaca informasi yang ditampilkan dengan nyaman

Informasi yang diberikan oleh responden kemudian diuji validitas serta reliabilitas setiap indikatornya dengan menggunakan bantuan perangkat lunak SPSS dan dilanjutkan dengan diolah secara deskriptif.

Hasil olah deskriptif kemudian digunakan untuk menentukan prioritas *user story* yang telah dipetakan. Selain itu, penulis juga menampung setiap keluhan dan temuan yang diberikan oleh responden sebagai *user story*.

Sprint Planning

Sprint planning dilakukan dengan menentukan jadwal pelaksanaan *sprint*, memilih *product backlog* yang akan dikerjakan pada *sprint* yang dijadwalkan, serta membuat daftar pekerjaan yang akan dikerjakan. satu item *product backlog* dapat dipecah atau dibagi menjadi beberapa daftar pekerjaan, namun tidak menutup kemungkinan bahwa lebih dari satu *product backlog* yang serupa digabung menjadi satu item *task*/pekerjaan.

Design Sprint

Design sprint dilakukan melalui empat fase pengembangan *prototype*, yaitu *Understanding*, *Diverge*, *Converge*, dan yang terakhir *Prototyping*. Fase *understanding* dilakukan dengan memahami permasalahan dan potensi perbaikan yang dapat dilakukan. Fase *diverge* dilakukan dengan mengumpulkan setiap kemungkinan solusi untuk diputuskan pada fase *converge*. Kemungkinan solusi yang terpilih pada fase *converge* kemudian dilanjutkan ke fase *prototype*.

Sprint Review

Sprint review dilakukan sebagai proses koreksi item *sprint backlog* yang telah dikerjakan. item *sprint backlog* yang tidak memerlukan koreksi diberi status *done*. Koreksi yang dianggap signifikan dapat dimasukkan kedalam *product backlog* untuk dikerjakan pada *sprint* selanjutnya.

Sprint Retrospective

Sprint retrospective dilakukan dengan membuat sebuah grafik yang terbagi dalam empat kuadran: hal apa saja yang berjalan dengan baik, hal apa saja yang dapat dipelajari lebih lanjut dan dapat diperbaiki, serta kejadian apa saja yang berjalan tidak sesuai rencana.

Evaluasi Akhir

Evaluasi Akhir dihitung dengan melakukan kembali evaluasi usability setelah *prototype* dibuat. Evaluasi akhir dilakukan dengan melihat rata-rata respon untuk melihat perkembangan dari desain lama dan desain baru. Pada evaluasi akhir, Validitas dan reliabilitas data tidak dihitung kembali karena kuisioner yang digunakan tidak berubah

ANALISIS DAN PEMBAHASAN

Identifikasi proses dan pengguna dilakukan dengan melakukan observasi. Pengguna yang terlibat dalam *planned maintenance system* antara lain: *Fleet Technical Manager*, *Technical Superintendent*, *Chief Engineer*, *Second Engineer/Engineers*, *Chief Officer*, *Second Officer/Officers*, serta *Fleet President*.

Fleet Technical Manager adalah manager yang bertanggung jawab atas seluruh proses PMS, baik ditinjau dari segi teknis dan biaya. *Technical Superintendent* adalah pengawas teknis yang bertanggung jawab atas seluruh proses PMS secara teknis. *Chief Engineer* adalah kepala kamar mesin/masinis 1, pimpinan di kapal yang bertanggung jawab untuk kelancaran proses PMS di bagian mesin kapal. *Engineer* adalah masinis 2, teknisi yang melaksanakan pemeriksaan di bagian mesin kapal. *Chief Officer* adalah pimpinan yang bertanggung jawab atas kelancaran proses PMS di bagian dek kapal. *Second Officer* adalah Mualim 2, Teknisi yang melaksanakan pemeriksaan di bagian dek kapal.

Aktivitas yang dilakukan didalam sistem antara lain: *Maintenance Scheduler*, *Standard Job*, *Request List*, *Part Request*, *Repair Request*, *Work Order*, dan *Work Order Report*.

Maintenance Schedule adalah jadwal pemeliharaan suatu alat. *Maintenance Scheduler* dibuat oleh seorang *Technical*

Superintendent dan di-review oleh seorang *Fleet manager*. *Standard Job* merupakan standar dalam melakukan suatu pekerjaan/tugas. Aktivitas ini dibuat oleh *Technical Super Intendent*, kemudian dilaksanakan oleh *Engineer/Officer*, dan di-review oleh *Chief Engineer/Chief Officer*. *Request List* adalah daftar permintaan barang yang ditujukan pada barang yang tersedia di kapal. *Request list* dibuat oleh seorang *officer/engineer*, kemudian di-review oleh seorang *Chief Officer/Chief Engineer*, dan disetujui oleh *Technical Super Intendent*. *Repair request* adalah permintaan perbaikan yang dibuat oleh *engineer/officer*, kemudian direview oleh *Technical Super Intendent*, dan disetujui oleh *Fleet Manager*. permintaan perbaikan kemudian dilanjutkan oleh *technical super intendent* untuk dilanjutkan menjadi *work order*. Setiap kegiatan *work order* harus dilaporkan. Laporan ini dibuat dalam bentuk *Work order report* oleh *Officer/engineer*, yang kemudian di review oleh *Chief Officer/Chief Engineer*, dan disetujui oleh *Technical Supert Intendent*.

Validitas Data

Penghitungan validitas data dilakukan dengan bantuan perangkat lunak SPSS, luaran yang dihasilkan mengungkap bahwa empat dari 26 indikator yang diuji validitasnya merupakan indikator yang tidak valid karena menghasilkan nilai p-value kurang dari *alpha* yang dikehendaki (0.05), variabel tersebut antara lain: p22, p32, p33, p34, dan p37.

Tabel 2. Validitas Indikator

ID	p-val	Ket	ID	p-val	Ket
P11	.000	Valid	P36	.000	Valid
P12	.002	Valid	P37	.153	invalid
P13	.000	Valid	P38	.003	Valid
P14	.012	Valid	P39	.000	Valid
P21	.004	Valid	P41	.003	Valid
P22	.244	invalid	P42	.000	Valid
P23	.001	Valid	P43	.043	Valid
P24	.000	Valid	P44	.007	Valid
P31	.003	Valid	P45	.001	Valid
P32	.110	invalid	P46	.000	Valid
P33	.103	invalid	P47	.003	Valid
P34	.000	Valid	P51	.002	Valid
P35	.000	Valid	P52	.011	Valid

Reliabilitas Data

Uji Reliabilitas data dilakukan dengan melihat tabel *reliability statistics* pada perangkat

lunak SPSS. Sebuah instrumen pengukuran dianggap reliabel jika menghasilkan nilai *Cronbach's Alpha* tidak kurang dari 0.700, nilai *Cronbach's Alpha* pada penelitian ini sebesar 0.794.

Tabel 3. Reliabilitas Data

Cronbach's Alpha	Cronbach's Alpha Based on Standarized Items	N of Items
.794	.798	26

Statistika Deskriptif

Empat puluh lima (78.9%) orang dari responden berjenis kelamin laki-laki, sedangkan sisanya sebanyak Dua belas (21.1%) orang dari responden berjenis kelamin perempuan. Jabatan terbanyak dalam penelitian ini adalah *Engineer* dan *Officer*, masing masing 17 dan 18 orang. *Chief Engineer*, *Chief Officer*, dan *Technical Super Intendent* sebanyak lima orang. Jumlah responden terbanyak berusia 36-44 tahun, dan responden dengan frekuensi paling sedikit berusia lebih dari 53 tahun sebanyak tiga orang, diikuti responden yang lebih muda dari 26 tahun sebanyak tujuh orang. Lebih dari setengah responden telah menggunakan sistem lebih dari lima bulan, sedangkan hanya delapan responden yang baru menggunakan sistem kurang dari satu bulan.

Product Backlog

Berdasarkan tabel frekuensi respon, *user story* yang teridentifikasi kemudian dirangkai dan dipetakan kedalam *product backlog* yang kemudian digunakan sebagai daftar yang harus dikerjakan dalam sebuah *sprint*. Detil *product backlog* dijelaskan pada tabel dibawah ini:

Tabel 4. *Product Backlog*

PBID	Story
PB1	Pengguna sulit mengenali dan mengingat menu yang ada pada sistem
PB3	Pemilihan warna belum mengingatkan user pada event yang terjadi pada sistem
PB4	Pengguna merasa kesulitan dalam membedakan antara teks warning dan bukan. Namun aktivitas yang dilakukan secara sukses dapat dibedakan.
PB9	tidak ada informasi yang mengindikasikan bahwa pengguna sukses masuk kedalam sistem

Sprint Iterasi-1

Setiap iterasi dilakukan dengan melakukan *sprint planning*, *Sprint*, *sprint review*, dan *Sprint retrospective*.

Sprint Planning

Sprint iterasi pertama dijadwalkan dimulai pada tanggal 05/11/2018 hingga 09/11/2018. *Sprint backlog* yang disusun antara lain:

Tabel 5. *Sprint Backlog*

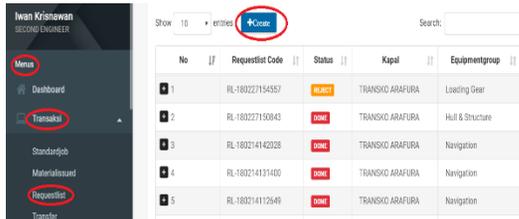
No	Task
	Pengguna sulit mengenali dan mengingat menu yang ada pada sistem (PB1)
PB1.1	Menyederhanakan model menu baru, mengurangi jumlah klik yang harus dilalui pengguna
PB1.2	Memperbaiki tata bahasa yang dipakai dalam menu
	Pemilihan warna belum mengingatkan user pada <i>event</i> yang terjadi pada sistem (PB3)
PB3.1	Menentukan palet warna yang akan digunakan pada Sistem saat terjadi sebuah <i>event</i>
PB3.2	Menentukan makna prioritas/konsekuensi masing masing warna
	Pengguna merasa kesulitan dalam membedakan antara teks warning dan bukan. Namun aktivitas yang dilakukan secara sukses dapat dibedakan. (PB4)
PB4.1	Membuat desain system alert/warning
	Tidak ada informasi yang mengindikasikan bahwa pengguna sukses masuk kedalam sistem (PB9)
PB9.1	Memperbaiki Login Page
PB9.2	Membuat header
PB9.3	Membuat footer

Design Sprint

Model menu yang saat ini diterapkan merupakan menu berupa *cascaded list* vertikal di sebelah kiri halaman, sebagai contoh jika seorang second engineer harus membuat sebuah *request list*, pengguna harus menekan tombol menu, transaksi, kemudian menu *request list* untuk masuk ke halaman tersebut. Tata bahasa yang digunakan dalam menu saat ini juga dinilai kurang tepat karena tidak menggunakan spasi.

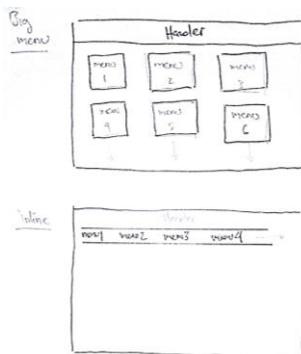
Perbaikan tata bahasa yang dipakai dalam menu dilakukan agar pengguna tidak perlu menebak apa yang dimaksud oleh sistem.

Tata Bahasa yang tidak baik berpotensi membuat pengguna kesulitan bernavigasi dalam sistem.



Gambar 4. Model menu saat ini

Fase *diverge* menghasilkan dua alternatif model menu yang akan digunakan dalam *prototype*. Tampilan berupa *big menu*, menu berupa link *inline* yang ditampilkan secara horizontal. Jumlah klik yang harus dilalui pada model menu saat ini sebanyak tiga klik, begitu juga dengan model tampilan *big menu*. Dengan menggunakan model *big menu*, pengguna harus menuju pada halaman *big menu* setiap kali harus berpindah halaman/aktivitas.



Gambar 5. Fase *diverge* tampilan menu

Pengguna cukup melalui satu kali klik pada model tampilan menu *inline* karena model menu ini selalu ditampilkan dibawah *header* aplikasi. Model menu yang dipilih pada fase *converge* adalah model tampilan menu *inline*. *Prototype* menu yang dibangun memungkinkan pengguna masuk dalam halaman yang diinginkan dengan melalui satu klik saja, selisih jumlah klik harus dilalui pengguna ini dipercaya dapat membantu mengurangi kompleksitas dalam mengingat menu yang harus ditekan oleh user.

Dengan model menu yang sederhana, pengguna diharapkan dapat lebih mudah mengingat menu mana yang harus ditekan. Dengan mengurangi jumlah klik yang harus dilalui, pengguna dapat berpindah dari satu halaman ke halaman lain dengan lebih cepat,

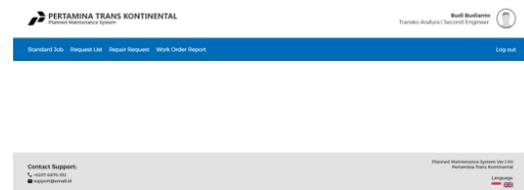
model menu yang kompleks atau bertingkat mengurangi kebutuhan pengguna untuk mengingat menu apa yang harus ditekan untuk sampai ke halaman tertentu.



Gambar 6. Header Saat ini

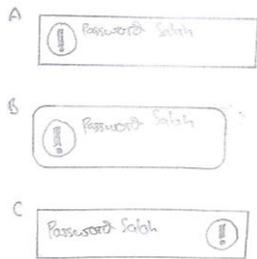
Header yang dibuat juga menampilkan informasi tentang sistem beserta informasi penggunaannya. Hal ini diharapkan dapat memberikan informasi bahwa pengguna telah sukses masuk kedalam sistem. Adanya informasi tentang judul halaman menjadi penting karena pengguna harus mengetahui dia sedang berada pada halaman apa sehingga tidak terjadi kesalahan dalam proses *input/update/delete*.

Informasi mengenai versi sistem yang sedang berjalan beserta kontak bantuan ditampilkan pada bagian footer sistem.



Gambar 7. *Prototype* Header, *inline* menu, dan Footer

Fase *diverge* dalam mengeksplorasi desain *alert/warning* menghasilkan beberapa alternatif. Desain *alert/warning* yang diajukan fokus pada penggunaan sudut *alert-box*. Desain *alert-box* A menggunakan sudut siku-siku tajam dan meletakkan icon disisi kiri kotak. Desain *alert-box* B menggunakan sudut melengkung (*rounded*). Desain *alert-box* C meletakkan icon disebelah kanan kotak dan menggunakan sudut siku-siku. Keputusan yang dipilih pada fase *converge* adalah menggunakan desain *alert-box* C. Sudut siku-siku tajam dipilih karena dinilai memberikan sifat tegas, sedangkan icon yang ditampilkan disebelah kanan memberikan informasi sekunder mengenai warning yang terjadi tanpa harus melewatkan teks/informasi utama.



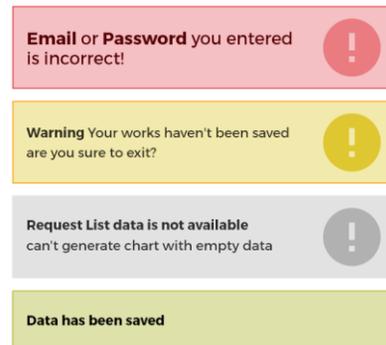
Gambar 8. Fase Diverge Alert-box

Warna yang digunakan pada *alert-box* juga mengikuti standar warna yang ditetapkan. Standar palet warna tersebut terdiri dari warna merah, kuning, hijau, dan abu-abu. Setiap warna juga memiliki makna terhadap konsekuensi yang muncul dari *alert-box*. *Alert-box* dengan warna merah menandakan bahwa peringatan yang muncul memiliki konsekuensi tinggi atau menandakan suatu kegagalan. Warna kuning memiliki makna konsekuensi yang lebih rendah dan dapat ditolelir. Warna hijau menandakan adanya aktivitas yang sukses, dan warna abu-abu digunakan sebagai info dan tidak memiliki konsekuensi.



Gambar 9. Palet warna Alert-box

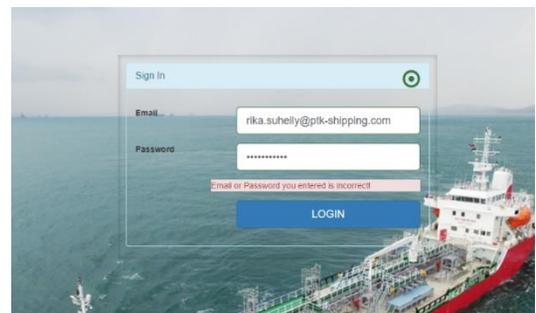
Palet warna *alert-box* juga digunakan dalam elemen sistem yang lain, sebagai contoh status/*state* sebuah pekerjaan. Pekerjaan *overdue* akan diberi warna merah, pekerjaan yang selesai akan selalu ditandai dengan warna hijau, pekerjaan *on progress* diberi warna biru.



Gambar 10. Prototipe Alert-box

Palet warna yang terstandarisasi memudahkan pengguna dalam mengartikan elemen UI sistem seperti *warning/alert* yang muncul. Ketika sistem memberikan *warning/alert* dengan warna yang tidak terstandar, pengguna berpotensi secara tidak sengaja mengabaikan kejadian tersebut.

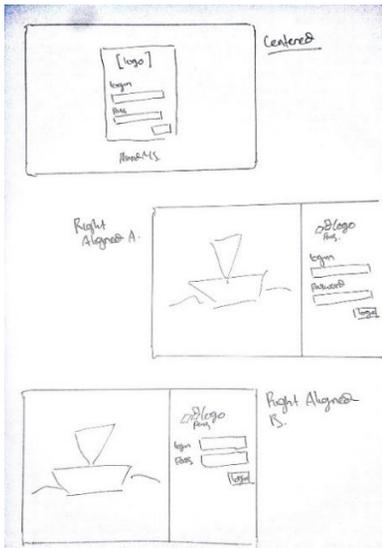
Desain halaman *login* saat ini tidak memberikan informasi kepada pengguna mengenai sistem, penulis juga menemukan beberapa inkonsistensi bahasa penggunaan kata *sign-in* dan *login* yang pada dasarnya memiliki makna yang sama. Selain itu pemilihan warna latar pada *form login* memiliki banyak ruang untuk perbaikan.



Gambar 11. Desain Halaman Login saat ini

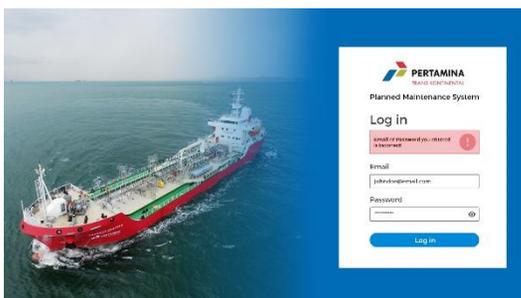
Fase *diverge* pada satuan pekerjaan membuat *login page* menghasilkan tiga alternatif desain. Desain pertama menampilkan form tepat ditengah layar. Desain ini mengadaptasi tampilan login yang diterapkan pada saat ini. Desain yang kedua dan ketiga menampilkan *form login* di sisi sebelah kanan layar. Yang membedakan antara desain kedua dan ketiga adalah letak *label form login*. Form login kedua meletakkan label secara linear vertikal, sedangkan form *login* ketiga meletakkan label dan *text-box* secara horizontal.

Tampilan desain halaman *login* yang dipilih pada fase *converge* dan dilanjutkan pada fase *prototype* adalah opsi desain kedua. *Prototype* yang dibuat memberikan informasi mengenai aplikasi yang sedang diakses oleh pengguna. Selain itu perhatian pengguna juga diarahkan pada teks “Log in” diawal *form* sehingga pengguna tau bahwa pengguna harus melakukan *login* dengan memasukkan *email* dan *password*.



Gambar 12. Fase *Diverge* Halaman *Login*

Alert-box ditampilkan setelah judul *form* diharapkan dapat mengambil perhatian pengguna bahwa telah terjadi kegagalan dalam memasuki sistem karena email dan password yang salah. Latar belakang putih pada *form login* yang baru diharapkan memberikan kontras warna yang baik.



Gambar 13. *Prototype* halaman *Login*

Sprint Review

Seluruh item *sprint backlog* pada *sprint* iterasi pertama selesai tepat waktu. Koreksi hanya dilakukan pada item PB3.1. palet warna sebelum *review* dinilai terlalu gelap. Koreksi

dilakukan ketika *sprint review* sehingga tidak perlu mengubah *product backlog*.



Gambar 14. Koreksi Palet Warna

Sprint Retrospective

Sprint retrospective pada iterasi pertama mengungkap bahwa penjadwalan dan alur pengerjaan tiap satuan tugas/*task* berjalan dengan baik, namun detil dalam pembahasan item *product backlog* dapat dikembangkan lebih lanjut.

<p><u>What went great?</u> Apa saja yang berjalan dengan baik?</p> <ul style="list-style-type: none"> • Penjadwalan • Alur pengerjaan 	<p><u>What can be learned?</u> Apa yang dapat dipelajari?</p> <ul style="list-style-type: none"> • -
<p><u>What is lacked?</u> Apa yang masih kurang?</p> <ul style="list-style-type: none"> • Detil dalam pembahasan item <i>product backlog</i> pada fase <i>understanding</i> 	<p><u>What went wrong?</u> Apa yang tidak sesuai rencana?</p> <ul style="list-style-type: none"> • -

Gambar 15. Kuadran *Sprint Retrospective*

Evaluasi Akhir

Evaluasi akhir dilakukan pada tanggal 26/11/2018 hingga 18/12/2018 setelah menunjukkan *prototype* yang dibuat dan mengukur kembali usabilitas kepada pengguna. Setiap variabel mengalami peningkatan rata-rata selain P13, P31, dan P35. P13 membahas

mengenai bahasa yang digunakan, menurun sebesar 0.75439. P31 membahas mengenai apakah tampilan mempercepat pekerjaan, menurun sebesar 0.01754. P35 membahas mengenai apakah pengguna dapat mengisi form dengan cepat, menurun sebesar 0.19298.

Rata-rata Aspek bahasa yang digunakan oleh sistem berkurang dapat diakibatkan oleh pengaruh kebiasaan bahasa yang dipakai oleh pengguna, begitu juga dengan tingkat kemudahan pengguna dalam menginputkan form, Pengguna juga dirasa memerlukan beberapa waktu untuk membiasakan diri dalam menggunakan sistem.

PENUTUP

Dari pengembangan desain *user interface* yang telah dilakukan, kesimpulan yang dapat ditarik antara lain: (1) Desain aplikasi yang baru mengadopsi model menu *horizontal* yang ditempatkan dibawah header aplikasi. (2) Desain aplikasi yang baru menstandarisasi palet warna untuk digunakan dalam elemen UI sistem. (3) Pemberitahuan dan peringatan divisualisasikan dalam bentuk *Alert-box* dengan sudut siku siku, icon mengenai *alert-box* ditampilkan di sisi kanan *box*.

DAFTAR PUSTAKA

- Banfield, R., Lombardo, T. C., & Wax, T. (2015). *Design Sprint: A Practical Guidebook for Building Great Digital Products*. Sebastopol: O'Reilly Media.
- Brien, J. O. (2008). *Comprehensive CMMS Implementation Guide*. Toronto: MA CMMS.
- Campbell, J. D., & James, R.-P. V. (2015). *Uptime: Strategies for Excellence in Maintenance Management*. Florida: CRC Press.
- CEN. (2010). *CEN - EN 13306:2010 Maintenance - Maintenance Terminology*. European Committee for Standardization.
- Conyer, M. (2008). User Usability Testing - How It Should be Undertaken? *Australian Journal of Education Technology*, 38-51.
- Frenkel, I., Lisnianski, A., & Khvaskin, L. (2009). Corrective Maintenance and Reliability Associated Cost Estimation of Aging Multistate System. *Computer Modelling and New Technologies*, 32-38.
- Ghasemifard, N., Shamsi, M., Rasouli Kenar, A., & Ahmadi, V. (2015). A New View at Usability Test Methods of Interfaces for Human Computer Interaction. *Global Journal of Computer Science and Technology: A Hardware & Computation*, 16-24.
- Ivory, M., & Hearst, M. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. *ACM Computing Surveys*, 470-516.
- Kullolli, I. (2008). Selecting a Computerized Maintenance Management System. *Clinical Engineering Management*, 276-278.
- Nadikattu, S. (2016). *Integrating User Experience (UX) Development with Agile Software Development Practices*. Karlskrona, Sweden: Blekinge Institute of Technology.
- Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Test*. Indianapolis: Wiley Publishing, Inc.
- Silva, T. S., Silveira, M. S., Maurer, F., & Hellmann, T. (2012). User Experience Design and Agile Development: From Theory to Practice. *Journal of Software Engineering and Application*, 743-751.
- Weichbroth, P., & Sikorski, M. (2015). User Interface Prototyping. Techniques, Methods, and Tools. *Studia Ekonomiczne. Zeszyty Naukowe*, 185-198.