

Implementasi Metode Scrum Dalam Pengembangan Test Engine Try Out Sertifikasi

Siti Saudah⁽¹⁾, Nia Oktaviani⁽²⁾, Muhammad Bunyamin⁽³⁾

^{1,2,3} Program Studi Sistem Informasi, Fakultas Ilmu Komputer

Jl. Ahmad Yani No 3 Seberang Ulu I Palembang, 30264

e-mail : ⁽¹⁾siti_saudah@binadarma.ac.id, ⁽²⁾niaoktaviani@binadarma.ac.id

⁽³⁾ muhammad.bunyamin@binadarma.ac.id

Abstract

Bina Darma University is one of higher education under the auspices of the Region II Higher Education Service Institute committed to producing graduates who have competence. One tangible form that can see from the activities carried out through the Faculty of Computer Science is the capacity-building and capabilities of prospective graduates through the certification program. Available certifications are competency certifications from world-renowned vendors that have licenses to issue certifications. But the problem that occurs is the failure caused by the unfamiliarity of participants in seeing the types of issues to be faced. Even though if it is seen to take a vendor certification exam, it will cost a lot of money and time is not small. For that reason, in this research, an alternative solution was made, namely by making a test engine try out certification. In the process of developing a test engine, the Scrum method is used, which focuses on speed in development. the results of the event show that the test engine produced can function properly and can be used as a try out the tool before conducting a certification exam.

Keywords : *Scrum, Test Engine, Competency Certification*

Abstrak

Universitas Bina Darma merupakan salah satu penyelenggara pendidikan tinggi di bawah naungan Lembaga Layanan Pendidikan Tinggi Wilayah II berkomitmen untuk menghasilkan lulusan yang memiliki kompetensi. Salah satu wujud nyata yaitu dapat dilihat dari kegiatan yang dilakukan melalui Fakultas Ilmu Komputer yaitu kegiatan peningkatan kapasitas dan kapabilitas calon lulusan melalui program sertifikasi. Sertifikasi yang tersedia merupakan sertifikasi kompetensi dari vendor ternama dunia yang memiliki lisensi untuk mengeluarkan sertifikasi. Namun permasalahan yang terjadi yaitu kegagalan yang disebabkan kurang terbiasanya peserta dalam melihat jenis soal-soal yang akan dihadapi. Padahal jika dilihat untuk mengikuti ujian sertifikasi vendor membutuhkan biaya yang mahal serta waktu yang tidak sedikit. Untuk itu didalam penelitian ini dilakukan upaya alternative solusi yaitu dengan dibuat test engine try out sertifikasi. Dalam proses pengembangan test engine digunakan metode Scrum yang berfokus pada kecepatan dalam pengembangan. hasil pengembangan menunjukkan bahwa test engine yang dihasilkan dapat berfungsi dengan baik serta dapat dijadikan sebagai alat try out sebelum dilakukannya ujian sertifikasi.

Kata Kunci : *Scrum, Test Engine, Sertifikasi Kompetensi*

1. PENDAHULUAN

Setiap lulusan dari pendidikan tinggi haruslah memiliki kompetensi yang sesuai dengan bidang studinya. Untuk itu setiap pendidikan tinggi berusaha untuk menghadirkan beragam cara agar terciptanya lulusan yang mempunyai agar memiliki kompetensi yang memadai. Beragam cara dapat dilakukan baik itu diselenggarakannya kegiatan workshop, training atau bahkan sertifikasi kompensi berskala nasional maupun internasional. Selain kompetensi yang harus dimiliki oleh lulusan yang melekat pada dirinya hal yang tak kalah penting adalah dokumen pembuktian bagi seseorang lulusan perguruan tinggi yaitu sertifikat kompetensi. Dokumen pendukung kompetensi diperlukan bagi seseorang lulusan perguruan tinggi sebagai penguatan atau bukti selain dari surat keterangan pendamping ijazah. Untuk mendorong terciptanya lulusan yang berkompeten perguruan tinggi sebagai penyelenggara pendidikan tinggi telah diatur sedemikian rupa oleh pemerintah melalui Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia salah satunya melalui instrument akreditasi. Akreditasi merupakan proses validasi dan verifikasi penyelenggara pendidikan tinggi untuk melihat *output* dan *outcome* yang dihasilkan.

Universitas Bina Darma merupakan salah satu penyelenggara pendidikan tinggi di bawah naungan Lembaga Layanan Pendidikan Tinggi Wilayah II berkomitmen untuk menghasilkan lulusan yang memiliki kompetensi. Salah satu wujud nyata yaitu dapat dilihat dari kegiatan yang dilakukan melalui Fakultas Ilmu Komputer yaitu kegiatan peningkatan kapasitas dan kapabilitas calon lulusan melalui program sertifikasi. Sertifikasi yang diselenggarakan diantaranya adalah Microsoft Technology Associate, SAP, NIIT, PASAS, CISCO, dan berbagai sertifikasi lainnya. Semua sertifikasi yang tersedia merupakan sertifikasi kompetensi dari vendor ternama dunia yang memiliki lisensi untuk mengeluarkan sertifikasi. Dalam proses penyelenggaraan sertifikasi tersebut dilakukan berbagai upaya seperti memasukkan materi sertifikasi pada matakuliah, training, botcamp, uji coba (*try out*) sebelum ujian sertifikasi dilakukan. Usaha tersebut merupakan proses untuk menghasilkan lulusan yang memiliki kompetensi yang mempunyai.

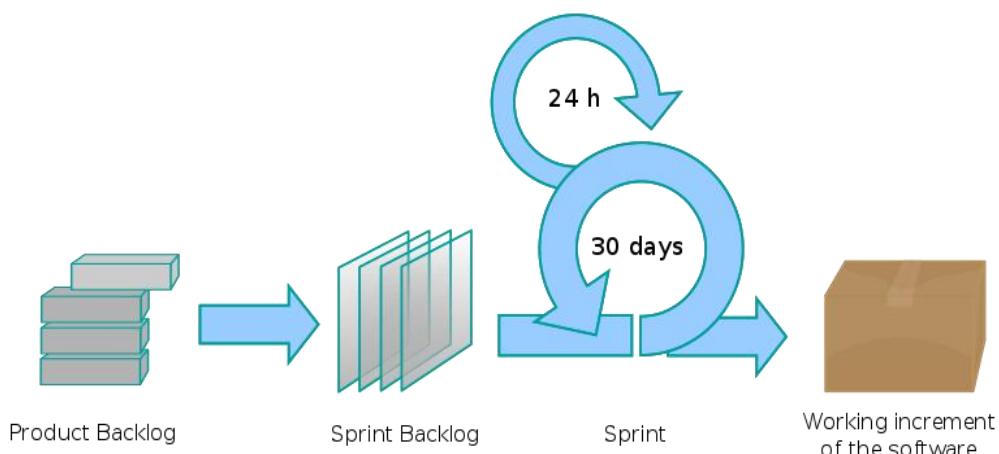
Untuk mendapatkan sertifikasi dari vendor tersebut sering kali calon lulusan yang mengikuti mengalami kegagalan. Kegagalan yang terjadi disebabkan kurang terbiasanya dalam melihat jenis soal-soal yang akan dihadapi. Padahal jika dilihat untuk mengikuti ujian sertifikasi vendor membutuhkan biaya yang mahal. Selain itu juga membutuhkan waktu yang tidak sedikit sebagai bahan persiapan untuk mengikuti ujian tersebut. Sesuai kondisi tersebut maka pihak pengelola ujian sertifikasi sebagai upaya peningkatan kompetensi perlu untuk membuat sebuah mekanisme agar calon peserta ujian sudah terbiasa dan sekaligus memahami jenis dan tipe soal ketika pelaksanaan ujian sertifikasi. Salah satu mekanisme yang dapat digunakan yaitu dibuatnya sebuah *test engine* yang digunakan oleh calon peserta untuk melakukan *try out* atau uji coba sebelum pelaksanaan ujian sertifikasi yang sebenarnya dimulai. Dengan adanya *test engine* untuk melakukan *try out* tersebut maka tingkat kegagalan peserta dalam ujian sertifikasi dapat dikurangi atau bahkan tidak ada. Pada akhirnya lulusan yang dihasilkan memiliki sertifikat kompetensi sebagai bentuk nyata pembuktian kemampuan alumni yang dihasilkan.

Sebagai upaya penyedian *test engine* yang digunakan untuk *try out* maka dalam proses pengembangannya digunakan metode pengembangan. Metode pengembangan terdapat berbagai jenis dan tipe baik metode classic ataupun modern. Metode pengembangan classic diantaranya waterfall, spiral, rapid application development, dan prototyping. Namun saat ini metode pengembangan yang sering digunakan adalah metode pengembangan modern seperti metode pengembangan *agile* (*agile methodology*). Dalam proses pengembangan modern dengan *agile* terdapat berbagai jenis metode yang dapat dipilih sebagai metode pengembangan yaitu: (1) *extreme programming*, (2) *adaptive software development*, (3) *agile modelling*, (4) *feature driven development*, (5) *dynamic system sevelopment methodology*, (6) *crystal*, dan (7) *Scrum* (Ibrahim, 2012). Kecepatan dalam proses pengembangan menjadi isu tersendiri dalam proses pengembangan. Dimana dari *agile methodology* proses pengembangan yang dapat dilakukan dengan cepat terdapat pada metode *Scrum* (Schön, Escalona, & Thomaschewski, 2015). Dengan demikian maka dalam menyediakan *test engine try out* digunakan *Scrum* sebagai metode pengembangan.

Proses pengembangan menggunakan metode *Scrum* terdapat lima tahapan pengembangan yaitu: (1) *backlog refinement*, (2) *sprint planning*, (3) *daily meeting*, (4) *reviews*, dan (5) *sprint retrospective*. Kelima proses pengembangan tersebut mengikuti tiga prinsip *Scrum* yaitu: *product owner (PO)*, *Scrum master (SM)*, dan *cross functional* (Alqudah & Razali, 2016). Metode *Scrum* dalam proses pengembangan sebuah proyek mengedepankan *sprint*, dimana kondisi ini telah terjadi ketika pertama kali Metode *Scrum* digunakan dalam proses pengembangan pada tahun 1990 (Robiansyah & Salma, 2017). *Sprint* didalam metode *Scrum* adalah proses pengembangan pada tiap tahapan. Dimana proses pengembangan di dalam *sprint Scrum* membutuhkan waktu yang sama untuk masing-masing *sprint* yaitu lebih kurang tiga puluh hari pengembangan. Sedangkan jenis dari *sprint* yang dilakukan pada proses pengembangan *Scrum* terdiri dari (1) *sprint planning*, (2) *daily scrum*, (3) *sprint review*, dan (4) *sprint retrospective* (Schwaber & Sutherland, 2012). *Scrum* juga memiliki kelebihan selain dari kecepatan, kelebihan tersebut yaitu dalam proses pengembangan selalu dilakukan pengecekan dan perubahan yang diperlukan sesuai dengan kebutuhan dan teknologi yang digunakan (Epandi, 2018). Keuntungan lain dengan metode *Scrum* yaitu dalam proses pengembangan dan pengujian sebuah proyek dapat dibuat berdasarkan modul sehingga fokus pengembangan dapat dilakukan (Meiliana, Bryan, Joshua, & Raymond, 2014).

2. METODE

Metode yang digunakan dalam penelitian ini merupakan metode pengembangan *Scrum* dengan proses pelaksanaan seperti yang diperlihatkan pada Gambar 1. *Scrum* adalah salah satu metode pengembangan turunan dari *Agile Development*. Dalam proses pengembangan terdapat kunci praktik (*key practices*) yang terdiri dari lima kunci. Kunci-kunci tersebut yaitu: (1) berpegang pada agenda jadwal, (2) bekerja konsisten sesuai *sprint*, (3) product backlog meruapakan tanda dalam pekerjaan, (4) produk backlog dasar melakukan sprint dan tim harus memutuskan produk dapat dikembangkan atau tidak, (5) SCRUM master bertanggung jawab menerima hasil sprint, (6) mengadakan *meeting* setiap hari kerja, dan (7) berfokus pada sprint, *meeting*, *review* dan *project timeline* (Kenett & Baker, 2010).



Gambar 1. Proses Pengembangan Scrum

A. Product backlog

Product backlog merupakan proses pengumpulan kebutuhan yang lakukan melalui pembuatan daftar kebutuhan. Selain dari daftar kebutuhan pada proses product backlog juga dibuat daftar (Subekti, Lukman, Indrawan, & Putra, 2014). Proses penggerakan yang dilakukan pada tahapan product backlog yaitu melakukan analisis terhadap kebutuhan *test engine* yang akan dikembangkan. Mengingat *test engine* ini akan digunakan oleh pihak fakultas Ilmu Komputer yang menaungi dua program studi yaitu program studi teknik informatika dan program studi sistem informasi maka kebutuhannya pun harus diakomodir dan disamakan dalam satu *test engine*. Proses pengumpulan kebutuhan tersebut dilakukan kepada pihak terkait yaitu pihak pengelolah sertifikasi yang ada di fakultas serta perwakilan program studi.

B. Sprint backlog

Sprint backlog adalah proses pemenuhan kebutuhan sesuai dengan yang diinginkan pada proses product backlog yang telah ditentukan (Dafitri & Elsera, 2017). Dari hasil analisis yang telah didapat maka kebutuhan *test engine* dapat diketahui maka *test engine* yang dibuat dengan tujuan utama yaitu memfasilitas calon alumni dalam hal ini calon peserta ujian sertifikasi untuk melakukan try out (ujicoba) soal-soal atau kisi-kisi sebelum dilakukannya ujian sertifikasi. *Test engine* yang dihasilkan diharapkan dapat melakukan (1) pendataan calon peserta ujian sertifikasi, (2) pengelolaan bank soal, (3) pengelolaan kelas *try out*, (4) pembuatan jadwal *try out*, (5) pelaksanaan *try out*, (6) koreksi atau hasil dari *try out*, dan (7) laporan nilai hasil *try out*.

C. Sprint,

Sprint merupakan proses dimana dilakukannya pemaparan hasil produk dalam bentuk prototype kepada pihak terkait dalam proses sertifikasi. Pihak terkait tersebut yaitu pengelola sertifikasi dan perwakilan program studi dilingkungan fakultas ilmu computer Universitas Bina Darma. Selain memaparkan terkait dengan produk prototype juga dilakukan pemaparkan berkaitan dengan hal teknis terkait peralatan yang dibutuhkan untuk menjalankan *test engine*.

D. Working Increment of the software

Working increment of the software merupakan tahapan pengembangan test engine sesuai dengan hasil sprint (*prototype*). Dalam proses pengembangan pada fase ini dilakukan penyesuaian kebutuhan dengan cara melakukan pertemuan untuk memberikan penyajian kepada pihak terkait yaitu pengelolah sertifikasi dan perwakilan program studi untuk mendapatkan masukkan. Setelah masukkan didapat maka selanjutnya dilakukan perbaikan dan kemudian dilakukan penyajian kembali kepada pihak terkait. Proses tersebut dilakukan berulang-ulang sampai dengan produk test engine dianggap telah memenuhi kebutuhan untuk melaksanakan *try out* untuk mengikuti ujian sertifikasi.

3. HASIL DAN PEMBAHASAN

Berdasarkan langkah-langkah atau proses pengembangan berdasarkan metode Scrum seperti yang diperlihatkan pada Gambar 1 maka hasil dari proses pengembangan dapat dijelaskan sebagai berikut.

3.1 Alur Proses Pelaksanaan *Try Out* pada *Test Engine*

Alur proses pelaksanaan *try out* merupakan alur bagaimana proses berjalannya pelaksanaan *try out* pada test engine yang dihasilkan. Pada Gambar 2 dapat dilihat alur proses pelaksanaan *try out* tersebut.

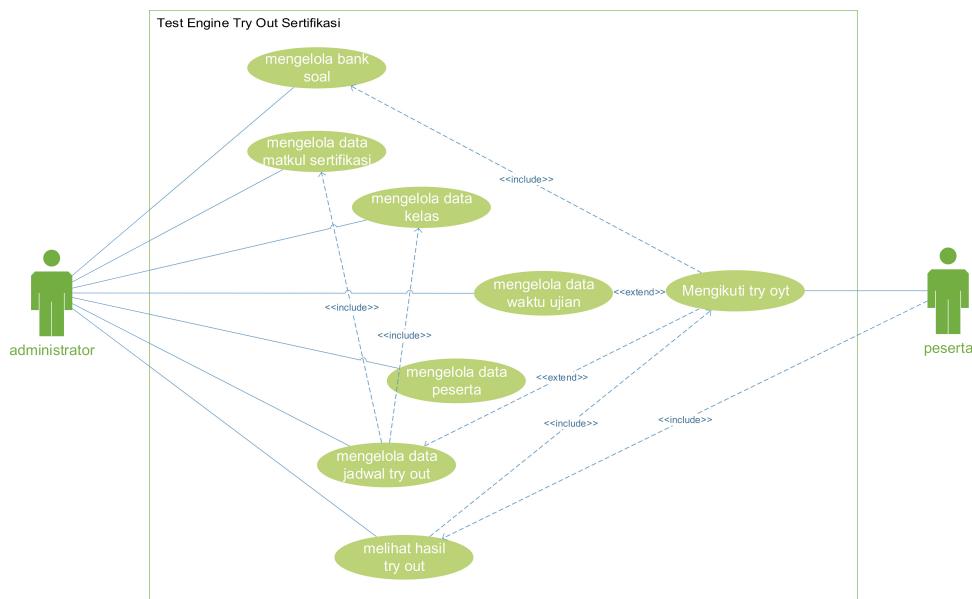


Gambar 2. Alur Proses Pelaksanaan *Try Out* Peserta

Alur proses pelaksanaan *try out* pada Gambar 2 merupakan proses pelaksanaan yang dilakukan oleh peserta. Sesuai yang diperlihatkan pada Gambar 2 bahwa proses pelaksanaan *try out* dilakukan dengan cara melakukan login dan kemudian barulah pelaksanaan *try out* dapat dilakukan. Namun jika peserta belum terdaftar maka harusnya melakukan pendaftaran terlebih dahulu. Setelah peserta melakukan *try out* maka otomatis nilai hasil *try out* diperlihatkan kepada peserta dan menjadi bahan evaluasi bagi peserta untuk mengikuti ujian sertifikasi.

3.2 Akses Pengguna *Test Engine*

Akses pengguna pada *test engine try out* sertifikasi yang dihasilkan terdapat dua jenis pengguna yaitu peserta *try out* dan *administrator test engine* itu sendiri. Pada Gambar 3 dapat dilihat merupakan akses pengguna untuk *test engine try out* sertifikasi yang dihasilkan.



Gambar 3. Use Case Diagram Akses Pengguna

Dari Gambar 3 dapat dilihat kedua akses pengguna memiliki keterhubungan satu sama lain. Dimana pengguna dengan akses peserta dapat melakukan atau mengikuti *try out* syarat harus dilakukan penjadwalan terlebih dahulu oleh *administrator*. Selain itu juga peserta untuk dapat mengikuti *try out* harulah terdaftar sebagai peserta dan dikelompokkan ke dalam sebuah kelas *try out*. Peserta juga dalam mengikuti *try out* harus mendapatkan soal dan waktu atau durasi *try out* sesuai dengan ketentuan yang telah dibuat oleh *administrator*. Pada Gambar 4 dapat dilihat halaman akses *administrator* pada *test engine try out*.



Gambar 4. Halaman Akses Administrator

Sesuai dengan akses pengguna seperti yang diperlihatkan pada Gambar 3 dan hasil implementasi untuk akses *administrator* pada Gambar 4 maka dapat dijelaskan fitur atau aktivitas yang dapat dilakukan administrator tersebut. Fitur yang dihasilkan diantanya adalah pengelolaan bank soal untuk sebagai instrumen ketika melakukan try out untuk pelaksanaan sertifikasi. Bank soal yang dimasukkan merupakan bank soal yang berasal dari materi pembelajaran dan soal-soal yang didapat dari vendor sebagai contoh soal. Selain itu juga terdapat soal-soal yang terdapat pada buku materi sertifikasi yang disediakan ketika melakukan workshop, training ataupun botcamp. Pada Gambar 5 dapat dilihat proses pengelolaan bank soal pada test engine try out sertifikasi yang dihasilkan.

FORM DATA SOAL

Matakuliah : Pilih Matakuliah
Bobot Soal : Pilih Bobot Soal

Soal :

Jawaban :

A. [Text Box] Benar B. [Text Box] Benar
 C. [Text Box] Benar D. [Text Box] Benar

Created By DSTI TIM © copyright 2019 Universitas Bina Darma

Gambar 5. Pengelolaan Bank Soal Try Out

Selain pengelolaan bank soal seperti yang diperlihatkan pada Gambar 5 hal yang tak kala pentingnya adalah pengelolaan jadwal try out sertifikasi. Untuk melakukan penjadwalan terdapat tahapan yang perlu dilakukan yaitu pembuatan kelas, penambahan peserta didalam kelas, waktu atau durasi try out dan baru pembuatan jadwal try out. Dalam proses pembuatan try out terdapat dua cara yang dapat dilakukan pertama dapat dilakukan berdasarkan kelas peserta dan matakuliah (sertifikasi) dan kedua berdasarkan peserta dan matakuliah (sertifikasi). Pada Gambar 6 dapat dilihat hasil penjadwalan yang dibuat pada test engine try out sertifikasi baik berdasarkan kelas dan matakuliah maupun peserta dan matakuliah.

DATA PESERTA TERJADWAL

No	NIM	Nama	Telepon	Status	Kelas	Batalikan
351	16142036P	WAWAN DIDIT M	081273915551	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
352	151420076	MUHAMMAD ARLANI	0855382881627	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
353	161420030	M. JULIANSYAH	082280403956	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
354	151420028	AGUNG WIJAYA	082103636278	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
355	17142038P	EFAN SAPITRA	082376963336	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
356	161420111	AGIT CAHYA NEGARA	081377739203	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
357	141420220	SOPIYAN EPENDI	082186869179	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
358	161420217	Bella Vista	-	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
359	161420022	AINUR RAHMAN	085215111441	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>
360	161420077	AKBAR YUSRI	081367739865	Terjadwal	MTA TI Genap 2019	<input type="button" value="Batalkan"/>

Created By DSTI TIM © copyright 2019 Universitas Bina Darma

Gambar 6. Hasil Penjadwalan Try Out

Setelah proses penjadwalan try out dilakukan seperti yang diperlihatkan pada Gambar 7 maka barulah peserta dapat melakukan try out pada login peserta. Untuk melakukan try out tersebut peserta harusnya melakukan login kedalam test engine dengan username dan password yang telah ditentukan. Ketika peserta telah melakukan login maka terdapat beberapa menu yang dapat akses diantanya adalah ujian sertifikas (try out) dan pengguna. Ujian sertifikasi merupakan menu untuk melakukan *try out* yang berisikan *try out* dan hasil *try out*. Sedangkan pengguna berisikan profil dan ubah *password* peserta. Pada Gambar 7 dapat dilihat merupakan halaman sebelum *try out* dimulai.



Gambar 7. Halaman Konfirmasi Try Out

Tampilan halaman konfirmasi seperti yang diperlihatkan pada Gambar 7 merupakan proses verifikasi atau konfirmasi yang perlu peserta lakukan. Kondisi tersebut dilakukan agar peserta melakukan *try out* untuk materi yang benar-benar ia pelajari untuk mengikuti ujian sertifikasi vendor (kompetensi). Setelah peserta mengamati dan membaca petunjuk *try out* maka peserta dapat mengklik link mulai ujian (*try out*) sesuai materi yang ia inginkan. Pada Gambar 8 dapat dilihat proses *try out* yang sedang berlangsung setelah peserta mengklik mulai ujian (*try out*). Dalam proses *try out* terdapat jumlah soal dan waktu pengerjaan *try out*, untuk menentukan kedua hal tersebut dilakukan pada administrator *test engine*. Selain kedua hal tersebut terdapat kondisi yang mempengaruhi hasil *try out* yaitu bobot soal. Bobot soal merupakan besaran bobot untuk masing-masing soal, dimana di dalam test engine *try out* ini setiap sial memungkinkan untuk memiliki bobot yang berbeda. Sehingga ketika peserta melakukan *try out* hal yang perlu diperhatikan juga adalah bobot soal. Pembobotan soal dibuat berdasarkan soal sertifikasi yang diujikan oleh vendor yang biasanya memiliki bobot yang berbeda walaupun tidak ditampilkan pada *interface* ketika peserta melakukan ujian.

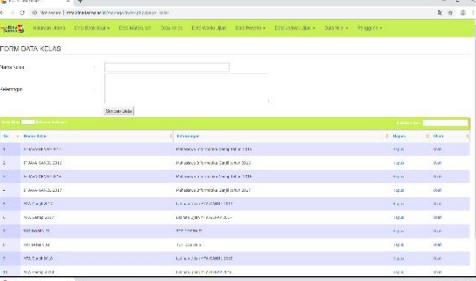
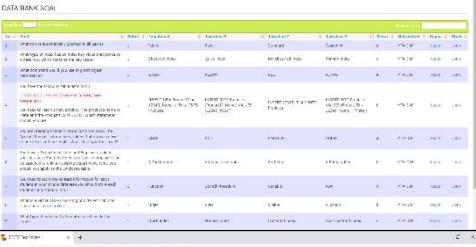
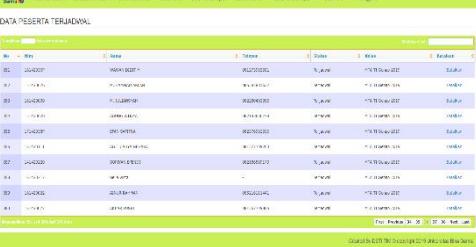


Gambar 8. Proses Try Out

3.3 Pengujian Test Engine

Pengujian test engine merupakan bentuk kepastian apakah test engine benar-benar berfungis atau tidak. Untuk itu dilakukan pengujian dengan teknik *black box*. Pengujian dengan teknik ini dilakukan untuk melihat fungsional test engine apakah berfungsi atau tidak. Pada Tabel 1 berikut dapat dilihat hasil pengujian yang dilakukan dengan teknik *black box*.

Tabel 1. Hasil Pengujian

Objek Uji	Pengamatan	Teknik Uji	Hasil Pengamatan
Pengelolaan Kelas		<i>Black box</i>	Berfungsi (Accepted)
Pengelolaan Matakuliah (Sertifikasi)		<i>Black box</i>	Berfungsi (Accepted)
Pengelolaan Waktu Ujian (<i>Try Out</i>)		<i>Black box</i>	Berfungsi (Accepted)
Pengelolaan Peserta		<i>Black box</i>	Berfungsi (Accepted)
Pengelolaan Bank Soal		<i>Black box</i>	Berfungsi (Accepted)
Pengelolaan Jadwal <i>Try Out</i>		<i>Black box</i>	Berfungsi (Accepted)

KESIMPULAN

Sesuai dengan uraian yang telah disampaikan dan dikemukakan sebelumnya maka kesimpulan yang dapat diambil berdasarkan implementasi metode Scrum dalam pengembangan test engine try out sertifikasi yang dibuat untuk Fakultas Ilmu Komputer Universitas Bina Darma sebagai upaya pemenuhan kebutuhan peningkatan kompetensi calon alumni yaitu sebagai berikut: (1) test engine yang dihasilkan dapat menjadi salah satu alternatif solusi bagi calon alumni dalam menghadapi ujian sertifikasi vendor yang rutin dilakukan pada Fakultas Ilmu Komputer Universitas Bina Darma. (2) test engine yang dihasilkan menunjukkan dapat melakukan try out (uji coba) dalam menghadapi ujian sertifikasi vendor dan berfungsi dengan baik sehingga layak untuk dijadikan alat bantu dalam menghadapi ujian sertifikasi yang akan dilakukan. (3) test engine yang dihasilkan telah dikembangkan sesuai dengan alur proses pengembangan dengan metode Scrum dan dinyatakan berfungsi dengan baik yang dibuktikan dari hasil pengujian menggunakan teknik *black box*.

DAFTAR PUSTAKA

- Alqudah, M., & Razali, R. (2016). A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering Information Technology*, 6(6), 828-837.
- Dafitri, H., & Elsera, M. (2017). Rancang Bangun Sistem Informasi Akademik Berbasis Web (studi kasus: SMA Swasta Harapan I Medan). *Query: Journal of Information Systems*, 1(2).
- Ependi, U. (2018). Implementasi Model Scrum pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang. *Jurnal Informatika: Jurnal Pengembangan IT*, 3(1), 49-55.
- Ibrahim, N. (2012). An Overview of Agile Software Development Methodology and Its Relevance to Software Engineering. *Jurnal Sistem Informasi*, 2(1).
- Kenett, R. S., & Baker, E. (2010). *Process improvement and CMMI for systems and software*: Auerbach Publications.
- Meiliana, M., Bryan, B., Joshua, F., & Raymond, R. (2014). Pengembangan Sistem Manajemen dan Analisis Key Performance Indicator "Smart Kpi" Berbasis Web. *ComTech: Computer, Mathematics Engineering Applications*, 5(2), 1119-1126.
- Robiansyah, H., & Salma, L. (2017). Sistem Informasi Mahasiswa Asing. *MATICS : Jurnal Ilmu Komputer dan Teknologi Informasi*, 9(1), 23-26.
- Schön, E.-M., Escalona, M. J., & Thomaschewski, J. (2015). Agile values and their implementation in practice. *International Journal of Interactive Multimedia and Artificial Intelligence*, 3(5), 61-66.
- Schwaber, K., & Sutherland, J. (2012). *Software in 30 days: how agile managers beat the odds, delight their customers, and leave competitors in the dust*: John Wiley & Sons.
- Subekti, M., Lukman, L., Indrawan, D., & Putra, G. (2014). Perancangan Case Tools untuk Diagram Use Case, Activity, dan Class untuk Permodelan Uml Berbasis Web Menggunakan HTML5 dan PHP. *ComTech: Computer, Mathematics Engineering Applications*, 5(2), 625-635.