

PENERAPAN *SCRUM* DALAM PENGEMBANGAN SISTEM *E-LEARNING* MODEL *CONTAINER* DENGAN *DOCKER* DAN *REACT JS*

Dimas Nugroho Putro¹, Muhamad Radzi Rathomi², Tekad Matulatan³
170155201005@student.umrah.ac.id

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji

Abstract

Currently, there are two important systems in the world of lectures, namely Academic Information Systems, and E-Learning. The two systems must be integrated so that students do not repeat the process so that it takes longer and can cause duplication of data. The purpose of this research is to apply Scrum in the Development of a Container Model E-Learning System with Docker and React JS. The method in this research is the Scrum framework which consists of several artifacts including the product backlog, sprint backlog, and increments. Scrum has structured and iterative stages, so that if the product produced in one sprint does not meet the needs, then in the next sprint a system can be developed according to user evaluation. The result of this research is an E-Learning System that is integrated with the Academic System of the Container model with Docker and React JS. This research was also tested by testing Blackbox Testing and User Acceptance Test. Based on the test results show that the built E-Learning System has met the requirements of user needs and functionality that has worked well.

Kata kunci: *scrum, e-learning, container, docker, react js*

I. Pendahuluan

Pesatnya perkembangan teknologi di era globalisasi ini berperan penting bagi dunia pendidikan, khususnya yang berkaitan dengan perkuliahan maupun administrasi perkuliahan, hal ini dikemukakan oleh Utomo dkk., (2018). Saat ini terdapat dua sistem penting dalam dunia perkuliahan, yaitu Sistem Informasi Akademik, dan *E-Learning*. Kedua sistem tersebut harus saling terintegrasi agar tidak terjadi duplikasi data yang menyebabkan kedua aplikasi menjadi kurang efektif. Terlebih ketika mahasiswa sudah mengisi Kartu Rencana Studi, mahasiswa harus mengambil kembali *course* pada sistem *E-Learning*. Hal ini mengakibatkan mahasiswa melakukan proses yang berulang-ulang sehingga memakan waktu lebih lama dan dapat menyebabkan adanya duplikasi data. Menurut Munir, (2009) istilah *E-Learning* lebih tepat ditujukan sebagai usaha untuk membuat transformasi proses pembelajaran yang ada di sekolah atau perguruan tinggi ke dalam bentuk digital yang dijumpai oleh teknologi internet. *E-Learning* yang peneliti kembangkan tidak mencakup fitur keseluruhan secara lengkap. Peneliti berfokus pada bagian materi mata kuliah, pengumpulan tugas, pengerjaan kuis, melihat nilai kuis yang sudah dikerjakan, diskusi kelas, dan melihat informasi tugas yang belum dikumpulkan, sehingga fitur yang peneliti buat hanya fitur inti dari sebuah *E-Learning*. *Role model* yang peneliti buat pada sistem *E-Learning* ini hanya sebagai mahasiswa. Pada penelitian ini peneliti menggunakan *Framework Scrum* berbasis *Agile Development* sebagai suatu kerangka kerja pengembangan sub-sistem *E-Learning*. Oleh karena itu, peneliti melakukan pengembangan sistem *E-Learning* model *container* dengan *Docker* dan *React JS* menggunakan *Framework Scrum*.

II. Metode Penelitian

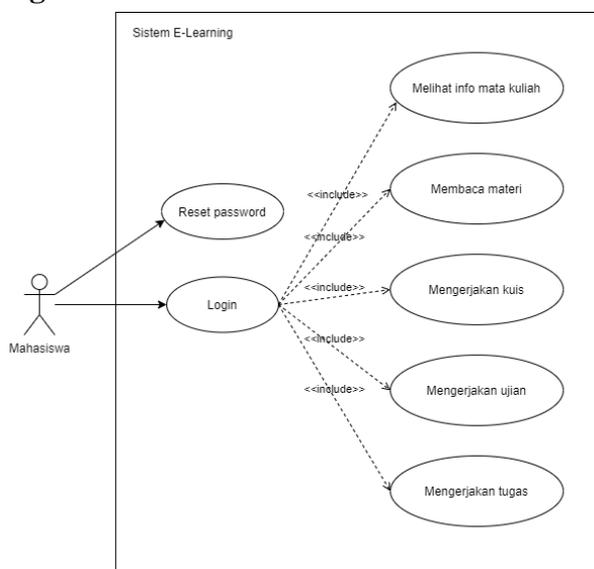
Pengumpulan Data

Data yang dibutuhkan dalam penelitian ini diperoleh melalui studi pustaka yaitu pengumpulan data yang bersifat teoritis dengan cara meninjau referensi yang didapatkan dari jurnal penelitian, buku, artikel, dan konferensi yang terkait dengan penerapan *Scrum* dan virtualisasi *container*. Untuk data komunikasi yang menghubungkan antara dua atau lebih perangkat lunak berupa API (*Application Programming Interface*) yang didapatkan dari *Product Owner*.

Analisis Data

Data yang dianalisis berupa *User Stories* yang kemudian dikembangkan menjadi fitur dan *business process* dalam sistem *E-Learning*. Analisis yang dianalisa berupa keterkaitan *user* dengan fitur dalam aplikasi dan seberapa penting kaitan antara kebutuhan *user* dengan aplikasi yang akan dibuat.

Perancangan Use Case Diagram



Gambar 1. Use Case Diagram

User Stories

User stories pada sistem *E-Learning* ditunjukkan pada Tabel 2.1.

Tabel 1. *User Stories*

No	Sebagai	Saya ingin	Sehingga
1	Mahasiswa	Melihat informasi mata kuliah	Mahasiswa mudah mengetahui pemberitahuan terbaru setiap membuka menu kuliah
2	Mahasiswa	Melihat dan membuka daftar materi dari setiap mata kuliah yang dipilih	Mahasiswa dapat melihat materi terbaru yang diunggah oleh Dosen
3	Mahasiswa	Membaca detail materi	Mahasiswa dapat membaca detail materi dari tiap mata kuliah
4	Mahasiswa	Melihat daftar kuis yang sedang berlangsung atau terlewati	Mahasiswa dapat melihat kuis yang diberikan oleh dosen

5	Mahasiswa	Mengerjakan kuis	Mahasiswa dapat mengerjakan dan memperoleh nilai dari kuis yang dikerjakan
---	-----------	------------------	--

Tabel 1. *User Stories* (Lanjutan)

No	Sebagai	Saya ingin	Sehingga
6	Mahasiswa	Melihat nilai kuis	Mahasiswa dapat melihat nilai kuis setelah ia selesai mengerjakan kuis
7	Mahasiswa	Melihat daftar tugas	Mahasiswa dapat mengetahui daftar tugas beserta statusnya
8	Mahasiswa	Mengirim hasil pengerjaan tugas dan mendownloadnya	Mahasiswa dapat mengirim tugas yang diberikan Dosen

III. Hasil dan Pembahasan

Adapun hasil implementasi *sprint* dari *sprint backlog* yang sudah dibuat saat *sprint planning* yaitu sebagai berikut:

Sprint I

Pada *sprint 1*, terdapat 14 *task* atau pekerjaan yang dikerjakan oleh *development team* Dimas Nugroho Putro dan M. Wahyu Irgan Agustino. Ruang lingkup *task* pada *sprint 1* ini hanya berupa perancangan desain antarmuka. *Sprint 1* dilaksanakan mulai dari tanggal 2 November 2020. *Sprint 1* memiliki total 16 *story point*. Untuk menghitung estimasi kerja pada *sprint 1* peneliti memakai nilai *focus factor* sebagai acuan untuk menghitung durasi *sprint* yang akan berlangsung. Nilai *focus factor* pada saat inisialisasi *sprint* yaitu bernilai 0.7 dikarenakan belum adanya nilai *focus factor* pada *sprint* sebelumnya. Setelah didapatkan nilai *focus factor* maka peneliti dapat mengetahui estimasi *work capacity ideal* yaitu 22,8. Untuk mengetahui estimasi waktu pengerjaan maka nilai *work capacity ideal* dibagi dengan jumlah *development team* yang berjumlah 2 orang yaitu 11,4. Maka estimasi waktu pengerjaan *sprint 1* dibulatkan menjadi 11 hari.

Sprint Review

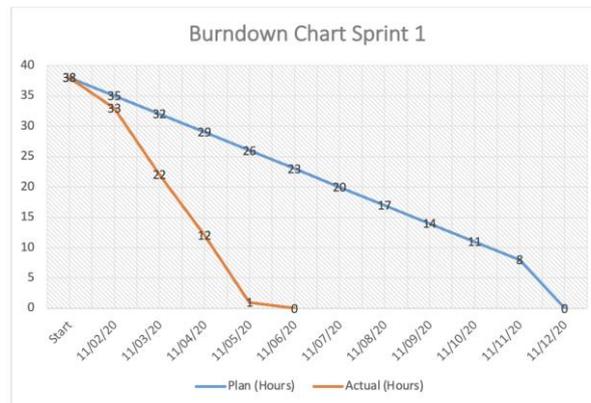
Sprint yang telah selesai dikerjakan akan dilakukan *review* oleh *Product Owner*. *Review* dari *sprint 2* yang telah dilakukan dipaparkan secara rinci pada penjelasan sebagai berikut.

- Konsep desain sudah bagus.
- Disarankan memakai *icon collection*.
- Peminjaman buku dan alat diganti dari *card* menjadi format *list*.
- Disarankan menguji *Quality of Service* pada perangkat lunak yang dikerjakan yaitu dengan menguji durasi *loading* pada tiap halaman.
- Diharuskan menggunakan *Unit testing* per tiap *sprint*.

Sprint Retrospective

Sebelum melanjutkan ke tahap *sprint 3*, perlu dilakukan kegiatan yaitu *sprint retrospective* sebagai evaluasi terhadap *task* yang telah dilakukan. Hasil dari *sprint retrospective* ini nantinya akan digunakan sebagai bahan evaluasi untuk pengerjaan *sprint* selanjutnya.

Burndown Chart



Gambar 2. Burndown Chart Sprint 1

Pada Gambar 2 terlihat bahwa *sprint* 1 dilakukan dalam 11 hari. Namun pada aktual pengerjaan *development team* dapat menyelesaikan seluruh *task* dalam 5 hari. Pada awal pelaksanaan *sprint*, terlihat bahwa terjadi penurunan drastis pada garis aktual pengerjaan. Hal ini menunjukkan bahwa tim *development* mengerjakan *task* diluar jadwal ketika *task* sebelumnya sudah selesai dikerjakan. Kondisi ini terjadi dikarenakan *task* pada *sprint* ini hanya berupa desain antarmuka sehingga tim *development* dapat menyelesaikan *task* lebih cepat dari perencanaan.

Team Retrospective

Hasil dari *team retrospective* pada *sprint* 1 dibagi menjadi 3 bagian yaitu :

a. Start

- Mengimplementasikan perubahan dari hasil *sprint review*.
- Kolaborasi tim secara *online* harus ditingkatkan lagi.
- Pertemuan dengan *product owner* harus lebih ditingkatkan lagi agar *quality control* terhadap sistem terjaga.
- Mulai menggunakan *tools* kolaborasi *online*.

b. Stop

- Mengerjakan *task* tidak tepat waktu.
- Menunda pertemuan dengan *product owner*.

c. Continue

- Melakukan *task* secara terstruktur.
- Estimasi pengerjaan *task* sudah baik.

Menghitung Focus Factor

Untuk menghitung estimasi durasi *sprint* 2, perlu dilakukan perhitungan *focus factor* dari *sprint* 1 yang telah dilakukan. Pada *sprint* 1 *development team* menggunakan 16 *story point* dan *work capacity* selama 5 hari, maka nilai *focus factor* pada *sprint* 1 adalah sebesar $\frac{(5 \times 2)}{16} \times 100 = 160\%$.

Sprint 2

Pada *sprint* 2, terdapat sepuluh *task* atau pekerjaan yang dikerjakan oleh *development team* Dimas Nugroho Putro dan M. Wahyu Irgan Agustino. Ruang lingkup *task* pada *sprint* 2 ini masih berfokus

pada perubahan perancangan desain antarmuka. Perubahan dilakukan pada saat melakukan hasil *sprint review* pada *sprint 1*. *Sprint 2* dilaksanakan mulai dari tanggal 28 Agustus 2021. *Sprint 2* memiliki total 20 *story point*. Untuk menghitung estimasi kerja pada *sprint 2* peneliti memakai nilai *focus factor* yang didapatkan dari *sprint* sebelumnya sebagai acuan untuk menghitung durasi *sprint 2* yang akan berlangsung. Nilai *focus factor* dari *sprint* sebelumnya adalah 160% atau 1,6. Setelah didapatkan nilai *focus factor* maka peneliti dapat mengetahui estimasi *work capacity ideal* yaitu 12,5. Maka estimasi waktu pengerjaan *sprint 2* dibulatkan menjadi 12 hari. Pengerjaan *sprint 2* dipaparkan lebih jelas pada penjelasan sebagai berikut.

Sprint Review

Sprint yang telah selesai dikerjakan akan dilakukan *review* oleh *Product Owner*. *Review* dari *Sprint 1* yang telah dilakukan dipaparkan secara rinci pada penjelasan sebagai berikut.

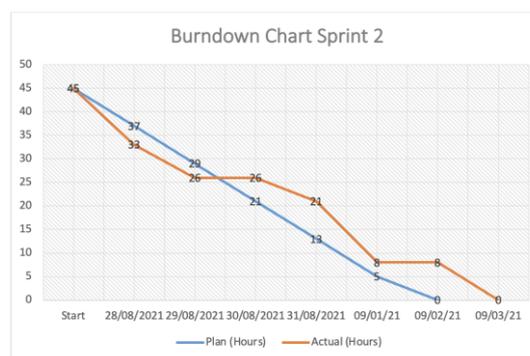
1. Gunakan palet warna yang lebih *soft* untuk komponen besar agar tidak sakit dimata.
2. Pada sistem *e-learning*, modul tidak dikelompokkan berdasarkan kategori melainkan disatukan semua.
3. Pada sistem akademik, table penjadwalan dihapus agar beban task menjadi berkurang.

Sprint Retrospective

Sebelum melanjutkan ke tahap *sprint 3*, perlu dilakukan kegiatan yaitu *sprint retrospective* sebagai evaluasi terhadap *task* yang telah dilakukan. Hasil dari *sprint retrospective* ini nantinya akan digunakan sebagai bahan evaluasi untuk pengerjaan *sprint* selanjutnya.

Burndown Chart

Sprint 2 dilakukan dengan 20 *story point* dengan durasi 6 hari. Untuk menggambarkan estimasi waktu pengerjaan dan waktu aktual pengerjaan, peneliti membuat *burndown chart* untuk memudahkan penggambaran analisis secara visual. Penggambaran *burndown chart* dari *sprint 2* dipaparkan pada Gambar 3 dibawah ini.



Gambar 3. *Burndown Chart Sprint 2*

Pada Gambar 3 terlihat bahwa *sprint 2* dilakukan dalam 6 hari. Namun pada aktual pengerjaan tim *development* menyelesaikan seluruh task dalam 7 hari. Pada 2 hari pertama pelaksanaan *sprint*, terlihat bahwa terjadi sedikit penurunan pada garis aktual pengerjaan. Hal ini menunjukkan bahwa *development team* mengerjakan lebih cepat dari perencanaan. Namun pada hari ke-4 sampai hari ke-7 terjadi keterlambatan pengerjaan. Kondisi ini disebabkan oleh ada faktor eksternal sehingga *development team* masih belum fokus dengan *task* yang telah diberikan.

Team Retrospective

Hasil dari *team retrospective* pada *sprint 2* dibagi menjadi 3 bagian yaitu :

a. Start

- Mengimplementasikan perubahan *design* dari hasil *sprint review*.

- Membuat ilustrasi untuk mempercantik *design*.
 - Mencari template baru atau memperbaiki *bug* pada *nicescroll*.
 - Mempoles *design* dengan mencari inspirasi dari *Learning Management System (LMS)* lain.
 - Kolaborasi tim secara *online* harus ditingkatkan lagi.
- b. *Stop*
- Mengerjakan *task* tidak tepat waktu.
 - Mengerjakan *task* terdahulu dan mulai mengerjakan *sprint* selanjutnya.
- c. *Continue*
- Melakukan *task* secara terstruktur.
 - Estimasi pengerjaan *task* sudah baik.

Menghitung *Focus Factor*

Untuk menghitung estimasi durasi *sprint* 3, perlu dilakukan perhitungan *focus factor* dari *sprint* 2 yang telah dilakukan. Pada *sprint* 2 *development team* menggunakan 20 *story point* dan *work capacity* selama 7 hari, maka nilai *focus factor* pada *sprint* 2 adalah sebesar $\frac{(7 \times 2)}{20} \times 100 = 140\%$.

Sprint 3

Pada *Sprint* 3, terdapat satu *task* atau pekerjaan yang dikerjakan oleh *development team* Dimas Nugroho Putro dan M. Wahyu Irgan Agustino, dan empat *task* yang dikerjakan sendiri oleh peneliti. *Sprint* 3 dilaksanakan mulai dari tanggal 9 September 2021. *Sprint* 3 memiliki total 14 *story point*. Untuk menghitung estimasi kerja pada *sprint* 3 peneliti memakai nilai *focus factor* yang didapatkan dari *sprint* sebelumnya sebagai acuan untuk menghitung durasi *sprint* 3 yang akan berlangsung. Nilai *focus factor* dari *sprint* sebelumnya adalah 140% atau 1,4. Setelah didapatkan nilai *focus factor* maka peneliti dapat mengetahui estimasi *work capacity ideal* yaitu 10. Maka estimasi waktu pengerjaan *sprint* 3 adalah 10 hari. Pengerjaan *sprint* 3 dipaparkan lebih jelas pada penjelasan sebagai berikut.

Sprint Review

Sprint yang telah selesai dikerjakan akan dilakukan *review* oleh *Product Owner*. *Review* dari *Sprint* 3 yang telah dilakukan dipaparkan secara rinci pada penjelasan sebagai berikut.

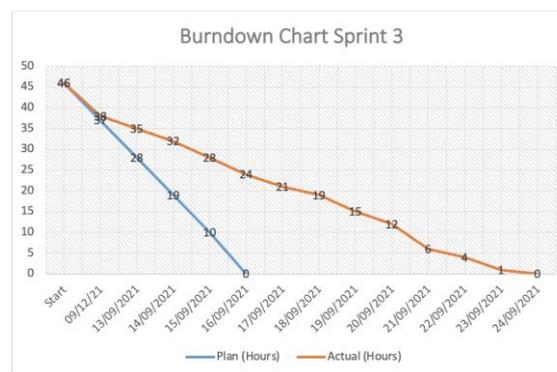
1. Konsep *design* sudah bagus walaupun belum desain *final*.
2. Beban *task development* tim dapat diberikan kepada anggota yang lain jika terlalu banyak.
3. Masih ditemukannya *bug* pada beberapa halaman dan fitur yaitu :
 - Ukuran *card* pada menu kuliah terlalu tinggi.
 - Logo pada *sidebar* dan *navbar* kadang tidak tampil.
 - Judul Materi pada *sidebar* khusus *course* masih *overlap*.
 - Pada saat *hover* menu pada *sidebar* masih ada animasi yang tidak mengikuti *rounded style*.
 - Jika menu pada *sidebar* memiliki *anak*, maka pada saat *anak* menu tersebut *dihover* muncul *state active* pada semua *anak* menu.
 - Foto profil kadang tidak muncul.
 - Pada rencana studi dibagian list rencana mata kuliah terdapat *bug checkbox* dapat tercentang walaupun belum ada mata kuliah yang dipilih.
 - SKS terambil belum terkalkulasi saat pengambilan mata kuliah.

Sprint Retrospective

Sebelum melanjutkan ke tahap *sprint* 4, perlu dilakukan kegiatan yaitu *sprint retrospective* sebagai evaluasi terhadap *task* yang telah dilakukan. Hasil dari *sprint retrospective* ini nantinya akan digunakan sebagai bahan evaluasi untuk pengerjaan *sprint* selanjutnya.

Burndown Chart

Untuk menggambarkan estimasi waktu pengerjaan dan waktu aktual pengerjaan, peneliti membuat *burndown chart* untuk memudahkan penggambaran analisis secara visual. Penggambaran *burndown chart* dari *sprint 3* dipaparkan pada Gambar 4.



Gambar 4. *Burndown Chart Sprint 3*

Pada Gambar 4 terlihat bahwa *sprint 3* dilakukan dalam 5 hari. Namun pada aktual pengerjaan *development team* menyelesaikan seluruh *task* dalam waktu 13 hari. Pada 2 hari pertama pelaksanaan *sprint*, terlihat bahwa *development team* mengerjakan *task* dengan tepat waktu, namun ketika memasuki hari ke-3 garis aktual pengerjaan menjauh dan berada di atas garis *planning* sampai hari ke-13. Hal ini menunjukkan bahwa terjadi pelaksanaan yang tidak sesuai dengan target. Kondisi ini disebabkan oleh faktor eksternal dari *development team* sehingga tidak fokus pada *task* yang telah diberikan.

Team Retrospective

Hasil dari *team retrospective* pada *sprint 3* dibagi menjadi 3 bagian yaitu :

a. *Start*

- Mulai masuk ke tahap *development*.
- Melakukan *daily scrum*.
- Mencari *service* untuk *deployment docker*.
- Membuat perencanaan *sprint* baru.
- Mulai buat dokumen UAT.
- Mulai mencari template dokumentasi kode.

b. *Stop*

- Mengerjakan *task* tidak tepat waktu.
- Mengerjakan *task* terdahulu dan mulai mengerjakan *sprint* selanjutnya

c. *Continue*

- Melakukan *task* secara terstruktur.
- Estimasi pengerjaan *task* sudah baik.
- Membuat pertemuan dengan *product owner* terkait ketersediaan data (API).

Menghitung Focus Factor

Untuk menghitung estimasi durasi *sprint* 4, perlu dilakukan perhitungan *focus factor* dari *sprint* 3 yang telah dilakukan. Pada *sprint* 3 *development* tim menggunakan 14 *story point* dan *work capacity* selama 13 hari, maka nilai *focus factor* pada *sprint* 3 adalah sebesar $\frac{(13 \times 2)}{14} \times 100 = 50\%$.

Sprint 4

Pada *Sprint* 4, terdapat enam *task* atau pekerjaan yang dikerjakan oleh *development team* Dimas Nugroho Putro dan empat *task* yang dikerjakan oleh M. Wahyu Irgan Agustino. *Sprint* 4 dilaksanakan mulai dari tanggal 2 Oktober 2021. *Sprint* 4 memiliki total 20 *story point*. Untuk menghitung estimasi kerja pada *sprint* 4 peneliti memakai nilai *focus factor* yang didapatkan dari *sprint* sebelumnya sebagai acuan untuk menghitung durasi *sprint* 4 yang akan berlangsung. Nilai *focus factor* dari *sprint* sebelumnya adalah 50% atau 0,5. Setelah didapatkan nilai *focus factor* maka peneliti dapat mengetahui estimasi *work capacity ideal* yaitu 20. Maka estimasi waktu pengerjaan *sprint* 4 adalah 20 hari. Pengerjaan *sprint* 4 dipaparkan lebih jelas pada penjelasan sebagai berikut.

Sprint Review

Sprint yang telah selesai dikerjakan akan dilakukan *review* oleh *Product Owner*. *Review* dari *Sprint* 4 yang telah dilakukan dipaparkan secara rinci pada penjelasan sebagai berikut

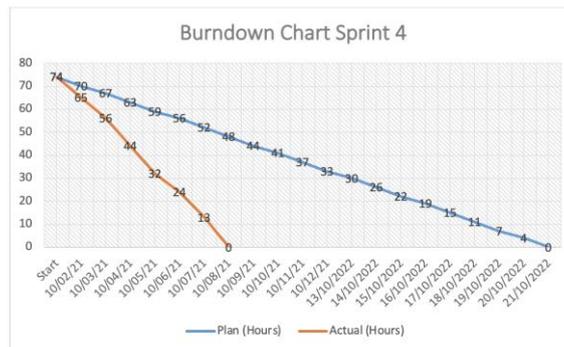
1. Konsep desain sudah baik
2. Masih ditemukannya beberapa *bug* di beberapa bagian :
 - Ukuran *card* pada menu kuliah terlalu tinggi.
 - Pada saat *hover menu* pada *sidebar* masih ada animasi yang *ngebug*.
 - Jika menu di *sidebar* memiliki *child*, maka pada saat *child* di *hover* muncul *state active* pada semua *child menu*.
 - Logo pada *sidebar* dan *navbar* kadang tidak tampil.
 - Foto profil terkadang tidak muncul.
 - Judul Materi pada *sidebar* khusus *course* masih *overlap*.
 - Pada rencana studi dibagian *list* rencana mata kuliah terdapat *bug checkbox* dapat tercentang walaupun belum ada mata kuliah yang dipilih.

Sprint Retrospective

Sebelum melanjutkan ke tahap *sprint* 5, perlu dilakukan kegiatan yaitu *sprint retrospective* sebagai evaluasi terhadap *task* yang telah dilakukan. Hasil dari *sprint retrospective* ini nantinya akan digunakan sebagai bahan evaluasi untuk pengerjaan *sprint* selanjutnya.

Burndown Chart

Untuk menggambarkan estimasi waktu pengerjaan dan waktu aktual pengerjaan, peneliti membuat *burndown chart* untuk memudahkan penggambaran analisis secara visual. Penggambaran *burndown chart* dari *sprint* 4 dipaparkan pada Gambar 5 dibawah ini.



Gambar 5. *Burndown Chart Sprint 4*

Pada Gambar 5 terlihat bahwa *sprint 4* dilakukan dalam 20 hari. Namun pada aktual pengerjaan tim *development* dapat menyelesaikan seluruh *task* dalam 7 hari. Pada ke-1 dan 2 pelaksanaan *sprint* terlihat tepat waktu dengan sedikit percepatan pengerjaan, namun pada hari ke-3 dan seterusnya terjadi penurunan garis aktual pengerjaan sampai hari ke-7. Hal ini menunjukkan bahwa tim *development* mengerjakan *task* lebih cepat dari yang perencanaan.

Team Retrospective

Hasil dari *team retrospective* pada *sprint 4* dibagi menjadi 3 bagian yaitu :

a. *Start*

- Mulai masuk ke tahap *development*.
- Melakukan *daily scrum*.
- Mencari *service* untuk *deployment docker*.
- Membuat perencanaan *sprint* baru.
- Mulai buat dokumen UAT.
- Mulai mencari template dokumentasi kode.

b. *Stop*

- Mengerjakan *task* tidak tepat waktu.
- Mengerjakan *task* terdahulu dan mulai mengerjakan *sprint* selanjutnya.

c. *Continue*

- Melakukan *task* secara terstruktur.
- Estimasi pengerjaan *task* sudah baik.
- Membuat pertemuan dengan *product owner* terkait ketersediaan data (API).

Menghitung Focus Factor

Untuk menghitung estimasi durasi *sprint 5*, perlu dilakukan perhitungan *focus factor* dari *sprint 4* yang telah dilakukan. Pada *sprint 4 development* tim menggunakan 20 *story point* dan *work capacity* selama 7 hari, maka nilai *focus factor* pada *sprint 4* adalah sebesar $\frac{(7 \times 2)}{20} \times 100 = 140\%$.

Sprint 5

Pada *Sprint 5*, terdapat tiga *task* atau pekerjaan yang dikerjakan oleh *development team* Dimas Nugroho Putro dan M. Wahyu Irgan Agustino. *Sprint 5* dilaksanakan mulai dari tanggal 23 Oktober 2021. *Sprint 5* memiliki total 26 *story point*. Untuk menghitung estimasi kerja pada *sprint 5* peneliti memakai nilai *focus factor* yang didapatkan dari *sprint* sebelumnya sebagai acuan untuk menghitung durasi *sprint 5* yang akan berlangsung. Nilai *focus factor* dari *sprint* sebelumnya adalah 140% atau 1,4%. Setelah didapatkan nilai *focus factor* maka peneliti dapat mengetahui estimasi *work capacity ideal* yaitu 9. Maka estimasi waktu pengerjaan *sprint 5* adalah 9 hari. Pengerjaan *sprint 5* dipaparkan lebih jelas pada penjelasan sebagai berikut.

Sprint Review

Sprint yang telah selesai dikerjakan akan dilakukan *review* oleh *Product Owner*. *Review* dari *Sprint 5* yang telah dilakukan memiliki beberapa poin yaitu:

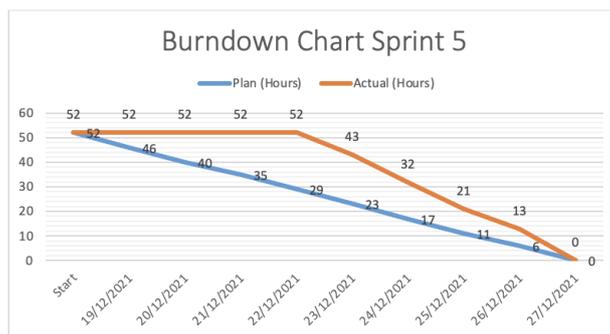
- Secara keseluruhan sistem sudah berjalan dengan baik
- Tampilan sudah menarik
- *Product owner* sudah menyatakan bahwa *product* memiliki standar *quality* yang baik, terlihat dari hasil *Unit Acceptance Testing* yang menunjukkan bahwa *presentase* responden terhadap aplikasi ini yaitu **Sangat baik**.

Sprint Retrospective

Sebelum melanjutkan menutup *sprint 5*, perlu dilakukan kegiatan yaitu *sprint retrospective* sebagai evaluasi terhadap *task* yang telah dilakukan. Hasil dari *sprint retrospective* ini nantinya akan digunakan sebagai bahan evaluasi jika ada *next development*.

Burndown Chart

Untuk menggambarkan estimasi waktu pengerjaan dan waktu aktual pengerjaan, peneliti membuat *burndown chart* untuk memudahkan penggambaran analisis secara visual. Penggambaran *burndown chart* dari *sprint 5* dipaparkan pada Gambar 6.



Gambar 6. *Burndown Chart Sprint 5*

Pada Gambar 6 terlihat bahwa *sprint 5* dilakukan dalam 9 hari. Pada aktual pengerjaan tim *development* dapat menyelesaikan seluruh *task* dalam hari ke-9. Pada hari ke-1 sampai ke-4, pelaksanaan *sprint* terlihat melakukan kemoloran jam kerja, namun pada hari ke-5 dan seterusnya terjadi penurunan garis aktual yang cukup lambat pengerjaan sampai hari ke-9. Hal ini menunjukkan bahwa tim *development* mmengerjakan *task* lebih lambat dari yang direncanakan.

Team Retrospective

Hasil dari *team retrospective* pada *sprint 5* dibagi menjadi 3 bagian yaitu :

- Start*
 - Melakukan pengujian UAT
- Stop*
 - Berhenti melakukan *sprint* plannin sampai ada *user story* baru (jika ada *next development*)
- Continue*
 - -

Menghitung *Focus Factor*

Perhitungan *focus factor* pada sprint 5 ini tidak bertujuan untuk menghitung estimasi durasi *sprint* berikutnya, namun digunakan untuk mencari nilai seberapa fokus *development team* mengerjakan *task* yang telah diberikan. Pada *sprint 5 development team* menggunakan 26 *story point* dan *work capacity* selama 5 hari, maka nilai *focus factor* pada sprint 5 adalah sebesar $\frac{(5 \times 2)}{26} \times 100 = 260\%$.

Pengujian Sistem dengan *Black Box*

Pengujian sistem *E-Learning* dengan metode *Black Box* dilakukan untuk menguji fungsionalitas sistem dengan skenario tertentu tanpa mengetahui struktur dari kode program. Berikut merupakan hasil pengujian yang peneliti lakukan pada sistem *E-Learning* yang dijelaskan pada Tabel 2.

Tabel 2. Pengujian Sistem dengan *Black Box*

No	Skenario	Hasil yang Diharapkan	Hasil	Kesimpulan
1	Melihat info mata kuliah	Tampil info mata kuliah yang sedang aktif	Tampil info mata kuliah yang sedang aktif	Berhasil (√)
2	Melihat <i>list</i> tugas	Tampil <i>list</i> tugas pada mata kuliah yang sedang aktif	Tampil <i>list</i> tugas pada mata kuliah yang sedang aktif	Berhasil (√)
3	Melihat <i>list</i> kuis	Tampil <i>list</i> kuis pada mata kuliah yang sedang aktif	Tampil <i>list</i> tugas pada mata kuliah yang sedang aktif	Berhasil (√)
4	Mencari Modul	Tampil <i>list</i> modul dengan hasil <i>keyword</i> yang dicari	Berhasil menampilkan <i>list</i> modul	Berhasil (√)
5	Membaca modul materi	Tampil modul materi pembelajaran	Berhasil menampilkan materi pada modul	Berhasil (√)
6	Mengerjakan kuis	Berhasil mengerjakan kuis dengan menjawab soal dan mengirim jawaban	Muncul <i>prompt</i> dan keterangan bahwa kuis sedang menunggu penilaian	Berhasil (√)
7	Mengerjakan kuis ketika melewati <i>deadline</i>	Tombol mulai kuis tidak tampil	Tombol mulai kuis tidak tampil, dan muncul keterangan tidak bisa mengerjakan kuis jika telah melewati masa enggang	Berhasil (√)
8	Mengerjakan kuis ketika modul belum dibuka	Tampil keterangan modul belum dibuka	Tampil keterangan modul belum dibuka	Berhasil (√)
9	Mengirim tugas	Tampil tugas yang telah dikirim	Tampil tugas yang telah dikirim	Berhasil (√)
10	Mengirim tugas ketika melewati <i>deadline</i>	<i>Form upload</i> tidak tampil, dan terdapat <i>alert</i> telah melewati masa <i>deadline</i>	<i>Form upload</i> tidak tampil, dan terdapat <i>alert</i> telah melewati masa <i>deadline</i>	Berhasil (√)

Pengujian Sistem dengan *User Acceptance Test*

Berdasarkan data hasil kuesioner tersebut, dapat dicari persentase dari masing-masing jawaban dengan cara membagi frekuensi jawaban dengan jumlah responden.

Tabel 3. Kuesioner UAT pada Bagian Umum

Variabel	Pertanyaan	Nilai				
		STS	TS	RR	S	SS
Desain	P1	0	0	0	4	2
	P2	0	0	1	4	1
	P3	0	0	1	4	1
	P4	0	0	0	5	1
	P5	0	0	2	3	1
	P6	0	0	2	4	0
	P7	0	0	1	4	1
	P8	0	0	3	2	1
Informasi	P9	0	0	2	0	4
	P10	0	0	0	5	1
	P11	0	0	2	4	0
Kecepatan Sistem	P12	0	0	0	0	6
Penyajian Informasi	P13	0	0	0	5	1
	P14	0	0	0	5	1
User Flow	P15	0	0	0	5	1
	P16	0	0	1	3	2
	P17	1	0	1	3	1
	P18	0	1	1	4	0
Kesimpulan Akhir	P19	0	0	0	5	1

Total	1	1	17	69	26
--------------	----------	----------	-----------	-----------	-----------

Hasil dari kuesioner yang diisi oleh *user* kemudian dihitung untuk mendapatkan total nilai, nilai tertinggi dan terendah. Berdasarkan dari hasil skor yang didapat, maka dapat dilakukan perhitungan untuk mendapatkan total nilai sebagai berikut:

Total responden menjawab SS	= 26 x 5	= 130
Total responden menjawab S	= 69 x 4	= 276
Total responden menjawab RR	= 17 x 3	= 51
Total responden menjawab TS	= 1 x 2	= 2
Total responden menjawab STS	= 1 x 1	= 1
Total nilai		= 460

Setelah mendapatkan total nilai, kemudian dapat dihitung nilai tertinggi dan terendah sebagai berikut:

1. Nilai tertinggi = $6 \times 19 \times 5 = 570$ (jika semua menjawab Sangat Setuju)
2. Nilai terendah = $6 \times 19 \times 1 = 114$ (jika semua menjawab Sangat Tidak Setuju)

Dari jumlah perhitungan diatas, nilai tertinggi yang didapatkan yaitu 570, nilai ini selanjutnya akan dihitung kembali untuk mencari nilai presentase sebagai berikut.

$$P = \frac{460}{570} \times 100\% = 80,7\%$$

Dari perhitungan persentase diatas, dapat ditarik kesimpulan bahwa sistem *E-Learning* yang telah dibuat oleh peneliti berdasarkan tingkat penerimaannya (*acceptance*) adalah Setuju. Hal ini dikemukakan oleh Priyatna (2020), yang mengatakan bahwa persentase yang diperoleh berkisar 61%-80% maka hasil pengujian tersebut dikatakan **Setuju**.

Pengujian Performa Sistem

Pengujian performa sistem dilakukan untuk mengetahui kualitas performa sistem yang telah dibangun. Pengujian ini menggunakan *tools Lighthouse*. Peneliti hanya menguji 3 parameter yaitu *Performance*, *Best Practice*, dan *SEO*. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4. Hasil Uji Performa Sistem

No	Halaman Uji	<i>Performance Score</i>	<i>Best Practice Score</i>	<i>SEO Score</i>
1	Info Mata Kuliah	70%	73%	100%
2	List Kuis	68%	73%	100%
3	List Tugas	65%	67%	100%
4	Silabus	74%	73%	100%
5	Modul Materi	59%	73%	100%
6	Modul Tugas	65%	67%	100%
7	Modul Kuis	68%	73%	100%
Rata-rata score		67%	71,2%	100%

Hasil dari pengujian yang ditampilkan pada tabel 4 menunjukkan bahwa peneliti melakukan uji performa pada 7 halaman. *Score* yang didapatkan pada tiap-tiap halaman memiliki nilai yang bervariasi. Namun pada *SEO score* seluruh halaman mendapatkan nilai yang sama yaitu 100%. Menurut hasil laporan dari *lighthouse* hal ini terjadi karena tiap-tiap halaman sudah mencapai *minimum requirement* untuk mesin pencari melakukan *indexing* alamat. Pada poin *performance score* memiliki nilai yang bervariasi pada tiap halaman, dan memiliki rata-rata *score* sebesar 67%. Sedangkan pada poin *Best Practice Score* mendapatkan nilai yang bervariasi pada tiap halaman, dan memiliki rata-rata *score* sebesar 71,2%.

IV. Kesimpulan

Dari hasil penelitian yang telah peneliti lakukan, dapat ditarik kesimpulan bahwa Penerapan *Scrum* dalam Pengembangan Sistem *E-Learning Model Container* dengan *Docker* dan *React JS* telah sukses diterapkan. Selama tahap pengembangan sistem *E-Learning* ini peneliti telah menghabiskan waktu selama 5 kali *sprint*. Dimana pada saat melakukan *sprint planning*, peneliti melakukan analisis *sprint backlog* untuk memberikan nilai *story point* berdasarkan skala prioritas dan kompleksitas dari setiap *task* dengan format bilangan *fibonacci*. Setelah mendapatkan total *story point*, maka dapat dilakukan perhitungan *focus factor* sebagai acuan untuk menghitung estimasi lama hari pengerjaan. Setelah ditetapkannya *sprint planning* maka tim *development* dapat memulai pengerjaan *task*. Ketika semua *task* selesai, maka tim *development* segera melakukan *sprint review* guna menjaga *quality control* dari sistem yang dibuat dan menerima masukan sebagai bahan pertimbangan untuk *sprint* selanjutnya. Dan terakhir adalah *sprint retrospective* sebagai waktu untuk mengidentifikasi bagaimana perasaan tim, apa yang dapat ditingkatkan kembali dari *sprint* yang telah selesai dilaksanakan. Peneliti beserta *development team* telah menyelesaikan seluruh *requirement* yang diberikan oleh *product owner* ditandai dengan seluruh *user stories* yang telah diimplementasikan dalam bentuk sistem *E-Learning* yang peneliti bangun. Dari hasil pengujian *User Acceptance Test* juga menyatakan bahwa pengguna Setuju dengan sistem yang peneliti bangun.

V. Daftar Pustaka

- Anwer, F., Aftab, S., Shah, S.S.M., dan Wahed, U., 2017, Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum, *International Journal of Computer Science and Telecommunications*, Vol. 8, No. 2, pp.1-7.
- Bella, M.R.M., Data, M., dan Yahya, W., 2019, Implementasi Load Balancing Server Web Berbasis Docker Swarm Berdasarkan Penggunaan Sumber Daya Memory Host, *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 3, No. 4, pp.3478-3487.
- Bik, M.F.R., dan Asmunin, 2017, Implementasi Docker Untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus: Jurusan Teknik Informatika UNESA), *Jurnal Manajemen Informatika*, Vol. 7, No. 2, pp.46-50.
- Cholifah, W.N., Yulianingsih, dan Sagita, S.M., 2018, Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android Dengan Teknologi Phoneyap, *Jurnal String*, Vol. 3, No. 2, pp.206-210.
- Hutrianto, dan Putra A., 2020, Implementasi Scrum Model Dalam Pengembangan Aplikasi Pelaporan Sampah Sebagai Wujud Smart Cleaning, *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, Vol. 5, No. 1, pp.9-19.

- Munir, 2009, *Pembelajaran Jarak Jauh Berbasis Teknologi Informasi dan Komunikasi*, Bandung: Alfabeta.
- Priyatna, B., Hananto, A.L., dan Nova, M., 2020, Application of UAT (User Acceptance Test) Evaluation Model in Minggon E-Meeting Software Development, *SYSTEMATICS*, Vol. 2, No. 3, pp.110-117.
- Ruseno, N., 2019, Implementasi Scrum Pada Pengembangan Aplikasi Sistem Reservasi Online Menggunakan PHP, *Jurnal Gerbang*, Vol. 9, No.1, pp.8-15.
- Schwaber, K., dan Sutherland, J., 2017, *The Scrum Guide*.
- Utomo, D.W., Alsyah, G.R., dan Prabowo, D.A., 2018, Integrasi Sistem Informasi Akademik dan E-Learning Berbais Web, *Jurnal Sistem Informasi dan Bisnis Cerdas (SIBC)*, Vol. 11, No. 1, pp.18-24.