

## Pengembangan Sistem Informasi *Display* Jadwal Perkuliahan dengan *Framework Laravel* Menggunakan Metode *Scrum*

Teuku Binzar Nawaf Musyaffa<sup>1</sup>, Muhammad Rizky Perdana<sup>2</sup>, Ahmad Rasyid Yusdi<sup>3</sup>, Ahlijati Nuraminah<sup>4</sup>

Program Studi Ilmu Komputer

Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ

[t.binzar.n@students.esqbs.ac.id](mailto:t.binzar.n@students.esqbs.ac.id)<sup>1</sup>, [m.rizky.p@students.esqbs.ac.id](mailto:m.rizky.p@students.esqbs.ac.id)<sup>2</sup>,  
[a.rasyid.y@students.esqbs.ac.id](mailto:a.rasyid.y@students.esqbs.ac.id)<sup>3</sup>, [ahlijati.nuraminah@esqbs.ac.id](mailto:ahlijati.nuraminah@esqbs.ac.id)<sup>4</sup>

**Abstrak.** Proses perkuliahan pada umumnya sudah dijadwalkan sejak awal semester. Pada kondisi tertentu dapat terjadi perubahan pada jadwal yang telah ditentukan. Perubahan jadwal seharusnya dapat diinformasikan dengan jelas kepada mahasiswa sehingga tidak menimbulkan masalah. Untuk mengatasi permasalahan ini, diperlukan suatu sistem untuk memberikan informasi secara cepat dan akurat ketika proses perkuliahan tersebut terganggu seperti berganti jadwal atau berpindah ruangan. Sistem ini diharapkan dapat diakses oleh mahasiswa pada tempat-tempat tertentu di area kampus yang mudah dijangkau. *Framework Laravel* dipilih karena cepat dan mudah dalam proses pengembangan. Selain itu digunakan metode *Scrum* di mana proses pengembangan dilakukan secara iteratif untuk menghasilkan sistem sesuai yang diharapkan pada setiap iterasinya. Hasil implementasi sistem menunjukkan bahwa dari 11 fitur yang telah ditentukan, 7 fitur berhasil diselesaikan dengan baik sesuai dengan kebutuhan, sedangkan 4 fitur lainnya akan dikembangkan pada rilis selanjutnya. Penggunaan metode *Scrum* dalam pengembangan sistem informasi *display* jadwal perkuliahan membantu tim menghasilkan sistem yang memenuhi kebutuhan pengguna. Dalam waktu pengembangan yang relatif singkat hanya selama 5 minggu, sistem sudah dapat diselesaikan dapat siap untuk digunakan.

**Kata Kunci:** Jadwal Kuliah, *Display* Informasi, *Scrum*, *Laravel*

### 1 Pendahuluan

Proses perkuliahan pada umumnya sudah dijadwalkan sejak awal semester. Namun pada pelaksanaannya sering terjadi perubahan karena berbagai hal, diantaranya dosen yang berhalangan hadir karena sedang sakit, atau mengikuti seminar pengembangan dosen, melakukan penelitian ataupun melaksanakan kegiatan pengabdian kepada masyarakat. Selain itu juga mungkin terjadi perubahan ruangan kuliah karena ruangan dipakai untuk aktivitas kegiatan kampus lainnya, atau ruangan sedang tidak dapat digunakan karena sedang mengalami perbaikan [1].

Kadang perubahan-perubahan seperti ini dapat terjadi dalam kondisi mendadak pada saat perkuliahan akan segera dimulai sesuai jadwal sehingga menimbulkan masalah. Mungkin saja terjadi mahasiswa sudah hadir sesuai jadwal pada ruang yang ditentukan, namun dosen belum hadir atau ada perubahan ruangan sehingga perkuliahan tidak dapat dilaksanakan tepat waktu [2]. Untuk mengatasi permasalahan ini, diperlukan suatu sistem untuk memberikan informasi secara cepat dan akurat ketika proses perkuliahan tersebut terganggu seperti berganti jadwal atau berpindah ruangan. Sistem informasi dikembangkan dalam bentuk web yang akan ditampilkan pada tempat-tempat strategis di area kampus menggunakan LED Monitor yang cukup besar. Sistem ini akan menampilkan jadwal kelas secara *real time* beserta perubahan-perubahan yang mungkin terjadi seperti perubahan jam, perpindahan ruangan, batal kelas, atau pergantian dosen.

Untuk membangun sistem yang dapat memberikan informasi perkuliahan secara *real time* dibutuhkan sebuah cara mengembangkan sistem yang dengan efisien, cepat dan sesuai dengan kebutuhan *user*. Untuk menghasilkan sistem dengan cepat, efisien dan berkualitas dapat dilakukan dengan memanfaatkan *framework* [3]. Saat ini banyak *framework* pembuatan web yang dapat dipilih untuk menghasilkan web dengan cepat, namun yang paling banyak digunakan saat ini adalah *Laravel*. *Laravel* banyak digunakan karena cukup mudah dalam penggunaannya, selain itu juga menawarkan banyak fitur tambahan yang dibutuhkan tanpa membuat dari awal.

Pengembangan sistem yang cepat juga harus disesuaikan dengan kebutuhan *user*. Perubahan kebutuhan *user* merupakan hal yang lumrah terjadi dan harus dapat diatasi dengan baik dalam pengembangan sistem. Agar dapat beradaptasi dengan perubahan kebutuhan *user* dengan baik, maka diperlukan metode pengembangan

sistem yang *sesuai*, salah satunya dengan melibatkan *client* dalam proses pengembangan [4]. Salah satu metode pengembangan sistem yang dapat beradaptasi dengan perubahan dengan baik adalah Scrum. Penerapan Scrum dengan melibatkan *client* diharapkan dapat meningkatkan fleksibilitas dan merangkul perubahan sedini mungkin sehingga sistem yang dihasilkan dapat sesuai dengan harapan [5].

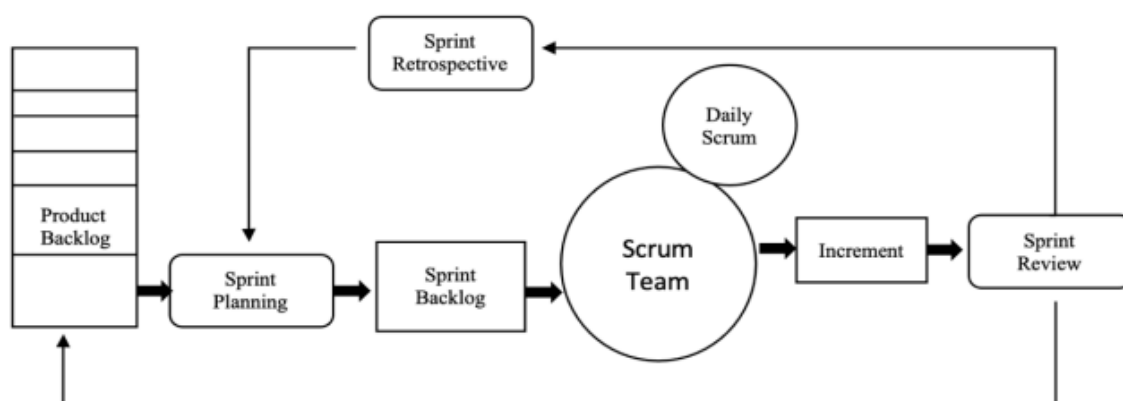
Dalam Scrum, terdapat beberapa agenda yang diselenggarakan untuk memfasilitasi tim dan *client* dalam proses pengembangan. Agenda Scrum yang dilakukan antara lain *sprint*, *sprint planning*, *daily scrum meeting*, *sprint reviews* dan *sprint retrospective* [6]. Kelebihan Scrum antara lain kecepatan menghasilkan *working product* dengan teratur [7]. Selain itu tahapan pengujian dilakukan di setiap unit fitur yang dikembangkan tanpa menunggu keseluruhan fitur selesai diimplementasikan [8].

Penelitian sebelumnya di Kampus UPI di Cibiru [1] mengembangkan sistem *display* jadwal kuliah berbasis web dengan menggunakan *framework* Laravel. Sistem ini diharapkan dapat menjadi solusi bagi dosen, mahasiswa, maupun stakeholder yang berkepentingan dalam proses penyelenggaraan akademik. Namun pendekatan yang digunakan dalam pengembangan sistem adalah metode *SDLC* (*Software Development Life Cycle*) dengan tahapan yang dilakukan adalah tahap analisis kebutuhan sistem, tahap perancangan sistem, tahap pembangunan sistem dan tahap pengujian. Setiap tahapan pengembangan sistem harus diselesaikan sebelum mulai tahap selanjutnya, sehingga jika terdapat kesalahan misalnya dari sisi desain, maka harus mengulang tahapan dari awal. Pendekatan ini dinilai kurang fleksibel dalam menerima perubahan kebutuhan dari sisi *user*. Berdasarkan kekurangan dari penelitian tersebut, maka diusulkan metode Scrum yang memiliki kelebihan dapat menerima perubahan dengan cepat, sehingga dapat memenuhi kebutuhan user dengan lebih baik.

Penelitian ini bertujuan untuk membangun sistem informasi display jadwal perkuliahan dengan menerapkan *framework* Laravel untuk mencapai kecepatan pengembangan dan metode Scrum untuk dapat menerima perubahan dengan lebih fleksibel.

## 2 Metode

Alur pengembangan sistem menggunakan metode Scrum seperti yang digambarkan pada Gambar 1.



**Gambar 1.** Alur pengembangan sistem menggunakan metode Scrum [9].

Metode Scrum yang ditampilkan pada Gambar 1 dijelaskan sebagai berikut

### 1. *Product Backlog*

*Product backlog* merupakan daftar kebutuhan sistem yang harus disusun menurut prioritas oleh *Product Owner* dan *Tim Development*. *Product Backlog* disusun pada awal mula proyek berlangsung. *Product Backlog* dibentuk dalam *user story* sederhana untuk memudahkan pengerjaan. *Product backlog* memuat informasi jenis *product backlog*, tingkat prioritas dan *story point*.

### 2. *Sprint Planning*

Setelah daftar *Product Backlog* ditentukan, kemudian dilaksanakan *sprint planning* untuk mengawali periode *sprint*. *Sprint planning* bertujuan untuk menentukan *Sprint Goal* yang ingin dicapai dan

*product backlog* apa yang akan diselesaikan selama durasi satu *sprint*. Masing-masing anggota tim *developer* memilih *product backlog item* yang akan dikerjakan. Pemilihan *product backlog item* ini berdasarkan pada tingkat kesulitan dan kemampuan *developer* dalam mengeksekusi fitur.

### 3. Sprint Backlog

Daftar *product backlog item* yang telah dipilih oleh tim *developer* pada *Sprint Planning* disebut *Sprint backlog*. *Sprint backlog* diharapkan dapat diselesaikan selama periode satu *sprint*. Setiap *sprint backlog* dipecah kembali menjadi beberapa *task* yang lebih kecil untuk mempermudah implementasi.

### 4. Scrum Team

Tim Scrum terdiri atas *Product Owner*, *Scrum Master* dan Tim *Developer*. *Product Owner* bertanggung jawab menentukan prioritas nilai bisnis dari produk yang dikembangkan. *Scrum Master* memiliki tugas mengarahkan tim untuk dapat menerapkan nilai-nilai Scrum dan memastikan setiap agenda Scrum dapat berlangsung dengan efisien. Tim *Developer* bertugas mengembangkan sistem sesuai urutan prioritas yang ditentukan *Product Owner*. Penelitian ini terdiri atas 1 orang *Product Owner*, satu orang *Scrum Master* dan 2 orang tim *developer*.

### 5. Daily Scrum

*Daily Scrum* diadakan setiap hari selama 15 menit per hari untuk memantau progres dari pengerjaan *sprint backlog* yang sudah ditentukan. Setiap anggota tim menjawab 3 pertanyaan berikut sebagai bentuk komitmen yaitu “Apa yang sudah dilakukan kemarin?”, “Apa yang akan dikerjakan hari ini?” dan “Kendala seperti apa yang sedang dihadapi?”.

### 6. Sprint Review

Pada hari terakhir dari periode *sprint*, tim akan melakukan *Sprint review* dengan mengundang *product owner*, *stakeholder* dan seluruh anggota *scrum*. *Sprint backlog* yang telah diselesaikan selama satu periode *sprint* akan didemonstrasikan oleh *Developer team*. *Product owner* menguji coba langsung fitur yang telah dikembangkan dan memberikan konfirmasi apakah fitur tersebut sudah sesuai atau masih memerlukan perbaikan lanjutan.

### 7. Sprint Retrospective

Setelah melakukan *Sprint Review*, anggota tim diberikan kesempatan kepada untuk refleksi terhadap *sprint* yang telah dilaksanakan, apa hal-hal baik yang dapat dipertahankan dan apa kekurangan yang dapat diperbaiki pada *Sprint* selanjutnya [10].

## 3 Hasil dan Pembahasan

### 3.1 Product Backlog

Daftar *product backlog* yang telah disusun berdasarkan prioritas dapat dilihat pada Tabel 1.

**Tabel 1.** Product Backlog

| No | <i>Product Backlog Item</i>   | Prioritas | <i>Story Point</i> |
|----|---|-----------|--------------------|
| 1  | Sebagai mahasiswa, saya ingin melihat jadwal kelas hari ini   | Tinggi    | 5                  |
| 2  | Sebagai operator saya ingin dapat mengubah sebuah sesi kelas supaya sesuai dengan kondisi yang ada real time. | Tinggi    | 8                  |
| 3  | Sebagai mahasiswa saya ingin melihat siapa dosen yang mengajar  | Tinggi    | 3                  |
| 4  | Sebagai mahasiswa saya ingin melihat informasi kelas yang batal   | Tinggi    | 3                  |
| 5  | Sebagai mahasiswa saya ingin melihat ruang di mana kelas saya berada  | Tinggi    | 3                  |
| 6  | Sebagai operator saya ingin dapat menginput jadwal setiap mata kuliah ke sistem.                              | Sedang    | 8                  |
| 7  | Sebagai mahasiswa, saya ingin melihat jadwal kelas hari ini dengan tampilan autoscroll                        | Sedang    | 5                  |
| 8  | Sebagai operator saya ingin status sebuah jadwal kelas berubah secara otomatis berdasarkan current time.      | Rendah    | 8                  |
| 9  | Sebagai operator saya ingin terautentikasi untuk mengakses menu-menu operator                                 | Rendah    | 5                  |
| 10 | Sebagai mahasiswa saya ingin menerima pemberitahuan via email jika ada perubahan pada jadwal                  | Rendah    | 8                  |
| 11 | Sebagai operator saya ingin dapat mengupload jadwal kuliah dalam bentuk                                       | Rendah    | 21                 |

| No | Product Backlog Item                        | Prioritas | Story Point |
|----|---|-----------|-------------|
|    | excel/csv agar tidak perlu menginput manual |           |             |

Dari sebelas product backlog yang telah ditentukan, Product owner dan tim developer menyepakati bahwa Product Backlog yang akan dikembangkan hanya yang memiliki prioritas tinggi dan sedang saja, product backlog dengan prioritas rendah akan dikerjakan pada pengembangan berikutnya

### 3.2 Sprint Backlog

Pada *Sprint Planning*, tim *developer* menentukan *product backlog item* yang akan dikerjakan pada *Sprint* tersebut. Product Backlog Item yang telah dipilih akan di-*breakdown* menjadi beberapa *task* yang lebih kecil untuk mempermudah setiap anggota tim dalam implementasi. Daftar sprint backlog yang dapat dilihat pada Tabel 2.

Tabel 2. Sprint Backlog

| No   | Sprint Backlog   | Story Point |
|--|--|-------------|
| <b>Sprint 1. Sprint Goal: Setup Project Environment</b>                    |  |             |
| 1  | Set up repository project  | 1           |
| 2  | Set up database menggunakan mysql  | 5           |
| 3  | Membuat front page   | 5           |
|  | <i>Estimasi Velocity</i>   | <b>11</b>   |
| <b>Sprint 2. Sprint Goal: Menampilkan jadwal kuliah hari dari database</b> |  |             |
| 1  | Sebagai mahasiswa saya ingin melihat jadwal kelas hari ini   | 5           |
| 2  | Membuat front page   | 5           |
| 3  | Input jadwal kuliah dan sesi sebanyak 16 sesi setiap mata kuliah                                   | 3           |
| 4  | Memperbaiki defect Sprint 1  | 1           |
|  | <i>Estimasi Velocity</i>   | <b>14</b>   |
| <b>Sprint 3. Sprint Goal: Input jadwal dan ubah sesi kelas</b>             |  |             |
| 1  | Sebagai operator saya ingin dapat mengubah sebuah sesi kelas supaya sesuai dengan kondisi yang ada | 8           |
| 2  | Sebagai operator saya ingin dapat menginput jadwal setiap mata kuliah ke sistem                    | 8           |
| 3  | Memperbaiki defect Sprint 2 berdasarkan feedback sprint 2  | 1           |
|  | <i>Estimasi Velocity</i>   | <b>17</b>   |
| <b>Sprint 4: Menampilkan data jadwal kuliah real time</b>                  |  |             |
| 1  | Sebagai mahasiswa saya ingin melihat siapa dosen yang mengajar                                     | 3           |
| 2  | Sebagai mahasiswa saya ingin melihat informasi kelas yang batal                                    | 3           |
| 3  | Sebagai mahasiswa saya ingin melihat ruang di mana kelas saya berada                               | 3           |
|  | <i>Estimasi Velocity</i>   | <b>9</b>    |

Pada *sprint* ke-5 tim *developer* menyelesaikan *product backlog item* yang belum tuntas dan melakukan sistem *testing* dari seluruh fitur yang sudah diselesaikan, kemudian melakukan proses *deployment* sistem.

### 3.3 Sprint Review

Dari Sprint Backlog yang telah ditentukan, penyelesaian setiap *task* pada *Sprint* bervariasi. Ada beberapa *backlog* yang dapat diselesaikan dalam satu *sprint*, namun ada beberapa *backlog* yang harus di-*carryover* ke *Sprint* berikutnya. Daftar *Sprint Backlog* yang telah diselesaikan pada setiap *Sprint* dapat dilihat pada Tabel 3.

Tabel 3. Sprint Review

| No   | Sprint Backlog   | Story Point | Status   |
|--|--|-------------|----------|
| <b>Sprint 1. Sprint Goal: Setup Project Environment</b>                    |  |             |          |
| 1  | Set up repository project  | 1           | DONE     |
| 2  | Set up database menggunakan mysql                                | 5           | DONE     |
| 3  | Membuat front page   | 5           | NOT DONE |
|  | <i>Actual Velocity</i>   | <b>6</b>    |          |
| <b>Sprint 2. Sprint Goal: Menampilkan jadwal kuliah hari dari database</b> |  |             |          |
| 1  | Sebagai mahasiswa saya ingin melihat jadwal kelas hari ini       | 5           | DONE     |
| 2  | Membuat front page   | 5           | DONE     |
| 3  | Input jadwal kuliah dan sesi sebanyak 16 sesi setiap mata kuliah | 3           | DONE     |

|  |  |           |           |
|--|--|-----------|-----------|
| 4  | Memperbaiki defect Sprint 1  | 1         | DONE      |
|  | <i>Actual Velocity</i>   | <b>14</b> |           |
| <b>Sprint 3. Sprint Goal: Input jadwal dan ubah sesi kelas</b> |  |           |           |
| 1  | Sebagai operator saya ingin dapat mengubah sebuah sesi kelas supaya sesuai dengan kondisi yang ada | 8         | DONE      |
| 2  | Sebagai operator saya ingin dapat menginput jadwal setiap mata kuliah ke sistem                    | 8         | DONE      |
| 3  | Memperbaiki defect Sprint 2 berdasarkan feedback sprint 2  | 2         | DONE      |
|  | <i>Actual Velocity</i>   | <b>17</b> | <b>18</b> |
| <b>Sprint 4: Menampilkan data jadwal kuliah real time</b>      |  |           |           |
| 1  | Sebagai mahasiswa saya ingin melihat siapa dosen yang mengajar                                     | 3         | DONE      |
| 2  | Sebagai mahasiswa saya ingin melihat informasi kelas yang batal                                    | 3         | DONE      |
| 3  | Sebagai mahasiswa saya ingin melihat ruang di mana kelas saya berada                               | 3         | DONE      |
|  | <i>Actual Velocity</i>   | <b>9</b>  |           |

### 3.4 Daily Scrum dan Sprint Retrospective

*Daily Scrum* dilakukan setiap hari pukul 13 siang secara daring melalui media Google Meet. Pada pelaksanaan *daily scrum* tidak selalu semua anggota tim hadir karena beberapa terkendala aktivitas lainnya pada saat yang sama. Namun secara umum *daily scrum* berjalan dengan lancar dan dapat memberikan dampak baik tim dalam hal koordinasi dan kolaborasi.

*Sprint retrospective* dilaksanakan setelah *Sprint Review Meeting*. *Sprint retrospective* dihadiri oleh *product owner*, *stakeholder*, *Scrum Master* dan tim *developer*. Secara umum pelaksanaan *Sprint retrospective* dapat memberikan masukan dan evaluasi bagi setiap anggota tim untuk memperbaiki performa pada *Sprint* berikutnya.

### 3.5 Rancangan Sistem

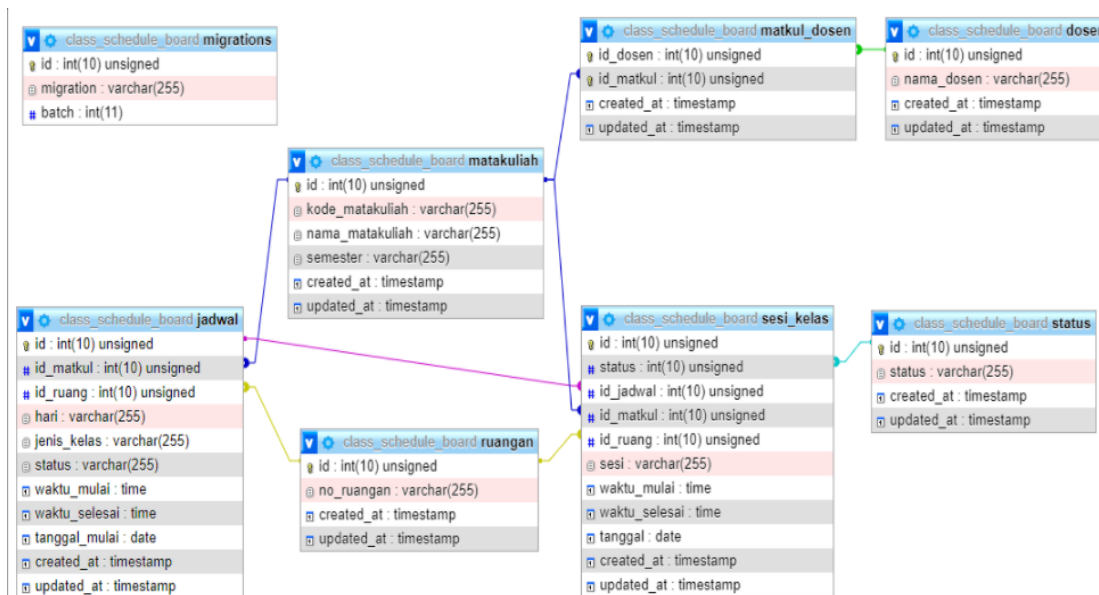
Sistem dirancang dengan tampilan sederhana untuk menonjolkan informasi utama yang dapat ditampilkan yaitu jadwal kuliah secara *real time*. Pada halaman awal desain terdapat data berupa tabel yang menampilkan jadwal kuliah pada hari tersebut dilengkapi jam, hari dan tanggal hari ini.

| Waktu                          | Kode    | Nama Mata Kuliah  | Dosen                                   | Semester | Ruang   | Status             |
|--------------------------------|---------|-------------------|---|----------|---------|--------------------|
| 07.30 - 09.10<br>09.40 - 11.20 | KBMT308 | Statistika R      | Danang Indrajaya /<br>Abdul Barir Hakim | 6, 8-BIS | R. 1901 | On Schedule        |
| 07.30 - 10.00                  | KKAI300 | Kecerdasan Buatan | Edo Surya Utama                         | 4-CS     | L. 1801 | Batal Kelas        |
| 07.30 - 10.00                  | KBSE402 | Manajemen Proyek  | Ahlijati Nuraminah                      | 6-CS     | R. 1804 | Sedang Berlangsung |
|                                |         |                   |   |          |         |                    |

Copyright 2022

**Gambar. 2.** Rancangan tampilan halaman utama sistem

Untuk keperluan pengolahan data, dirancang sebuah basis data yang terdiri atas beberapa tabel untuk menyimpan data-data yang dibutuhkan, seperti data dosen, data mata kuliah, data ruangan, data mata kuliah yang diampu dosen, data jadwal mata kuliah dan data sesi mata kuliah. Sebuah mata kuliah mungkin diampu oleh lebih dari satu dosen dan memiliki lebih dari satu sesi (misalnya sesi teori dan sesi praktek) yang juga menggunakan ruangan berbeda (misalnya sesi teori di ruang kelas, sesi praktek di laboratorium). Rancangan basis data yang digunakan dapat dilihat pada Gambar 3.



Gambar. 3. Rancangan basis data sistem

### 3.6 Arsitektur Sistem

Sistem dibangun menggunakan framework Laravel 8 dengan bahasa pemrograman PHP dan basis data MySQL. Halaman-halaman yang diimplementasikan antara lain: halaman utama berisi data jadwal mata kuliah *real time*, halaman input jadwal bagi operator, *form insert* kelas, sesi kelas dan *insert* sesi kelas bagi operator. Semua halaman dikembangkan menggunakan arsitektur MVC pada Laravel yaitu pada *blade*. *Blade* merupakan fitur yang telah disediakan oleh Laravel untuk proses *templating* sederhana yang digunakan dalam proses pengembangan tampilan halaman web. Daftar *file* menurut arsitektur MVC sistem dapat dilihat pada Tabel 4.

Tabel 4. Daftar file MVC

| No                | Nama File               | Deskripsi  |
|-------------------|-------------------------|--|
| <b>Model</b>      |                         |  |
| 1                 | Dosen.php               | Berisi class Dosen dari tabel dosen                                      |
| 2                 | Jadwal.php              | Berisi class Jadwal dari tabel jadwal                                    |
| 3                 | MataKuliah.php          | Berisi class MataKuliah dari tabel matakuliah                            |
| 4                 | Ruangan.php             | Berisi class Ruangan dari tabel ruangan                                  |
| 5                 | SesiKelas.php           | Berisi class SesiKelas dari tabel sesi_kelas                             |
| 6                 | Status.php              | Berisi class Status dari tabel status                                    |
| 7                 | User.php                | Berisi class user pengguna sistem  |
| <b>View</b>       |                         |  |
| 1                 | classSchedule.blade.php | Berisi template halaman utama sistem yang berisi tabel jadwal kuliah     |
| 2                 | formInsertKelas.php     | Berisi template halaman untuk mengisi data kelas                         |
| 3                 | formSesiKelas.blade.php | Berisi template halaman untuk mengisi data sesi kelas                    |
| 4                 | sesiTabel.blade.php     | Berisi template halaman yang menampilkan tabel sesi kelas                |
| <b>Controller</b> |                         |  |
| 1                 | JadwalController.php    | Berisi controller untuk mengatur tampilan dan data terkait jadwal kuliah |
| 2                 | SesiKelasController.php | Berisi controller untuk mengatur tampilan dan data terkait sesi kelas    |

## 4 Pembahasan

### 4.1 Hasil Implementasi Sistem

Setelah melalui iterasi *Sprint* sebanyak 4 kali, sistem yang diharapkan dapat diselesaikan dengan baik. Halaman-halaman fitur yang selesai diimplementasikan antara lain halaman utama yang menampilkan jadwal kuliah *real time*, halaman daftar jadwal bagi operator, halaman input jadwal kelas dan sesi kelas bagi operator. Tampilan halaman utama sistem dapat dilihat pada Gambar 4.



The screenshot shows the main page of the ESQ Business School system. At the top left is the school's logo. The main heading reads "SELAMAT DATANG DI ESQ BUSINESS SCHOOL". To the right, a date and time widget displays "Sabtu 2021-06-19" and "19:51:20". Below this is a search bar and a table of class entries. The table has columns for Waktu, Kode, Nama Matakuliah, Dosen, Semester, Jenis Kelas, Ruang, and Status. Two entries are visible: one for Multimedia Animasi and another for Google Cloud Computing Foundations.

| Waktu               | Kode    | Nama Matakuliah                    | Dosen                      | Semester  | Jenis Kelas | Ruang  | Status      |
|---------------------|---------|------------------------------------|----------------------------|-----------|-------------|--------|-------------|
| 16:00:00 - 18:00:00 | KBMM102 | Multimedia Animasi                 | Johny Hizkia Siringo Ringo | 2-BIS, CS | Lab         | L.1801 | On Schedule |
| 07:00:00 - 10:00:00 | KBGB302 | Google Cloud Computing Foundations | Ahljati Nuraminah          | 6-CS      | Teori       | R.1803 | On Schedule |

**Gambar. 4.** Tampilan halaman utama sistem

Pada halaman ini terdapat sebuah logo dan judul halaman yang menunjukkan bahwa pengunjung telah masuk ke situs awal web ini, pada bagian kanan atas terdapat *input search* yang berguna mencari sesi kelas yang ingin diketahui, dan tabel yang berisi semua data sesi kelas yang ada pada hari tersebut. Pada bagian paling atas kanan terdapat jam *widget* yang terus *ter-update* per detiknya, serta hari dan tanggal.

Halaman sesi kelas digunakan oleh operator untuk mengubah jadwal sebuah sesi kelas. Halaman ini menampilkan semua data sesi kelas dan sebuah tombol *action edit* yang berguna untuk meng-*update record* pada masing-masing sesi dengan tampilan seperti pada Gambar 5.



The screenshot shows the class session list page. It features the same header as Gambar 4. Below the search bar is a table with columns for Tanggal, Waktu, Kode, Nama Mata Kuliah, Dosen, Semester, Jenis Kelas, Sesi Kelas, Ruang, Status, and Action. The table contains six rows of session data, each with an "Edit" button in the Action column.

| Tanggal    | Waktu               | Kode    | Nama Mata Kuliah | Dosen                              | Semester | Jenis Kelas | Sesi Kelas | Ruang  | Status  | Action |
|------------|---------------------|---------|------------------|------------------------------------|----------|-------------|------------|--------|---------|--------|
| 2021-02-22 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 1          | RAUDI  | Selesai | Edit   |
| 2021-03-01 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 2          | L.1801 | Selesai | Edit   |
| 2021-03-08 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 3          | L.1801 | Selesai | Edit   |
| 2021-03-15 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 4          | L.1801 | Selesai | Edit   |
| 2021-03-22 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 5          | L.1801 | Selesai | Edit   |
| 2021-03-29 | 07:30:00 - 09:10:00 | KBMT308 | Statistika R     | Danang Indrajaya/Abdul Barir Hakim | 6, 8-BIS | TEORI       | 6          | L.1801 | Selesai | Edit   |

**Gambar. 5.** Tampilan halaman daftar sesi kelas

Jika tombol *edit* ditekan maka tombol tersebut akan mengakses *url* pada halaman *form* ubah sesi dan melempar data sesi kelas sesuai dengan *record* yang dipilih.

### 4.2 Pembahasan Sistem

Pengujian sistem dilakukan menggunakan teknik *black-box testing* pada fungsionalitas sistem yang telah dibangun. Pengujian dilakukan secara bertahap sejak implementasi di masing-masing *sprint* yaitu dengan melakukan *unit testing*. Pada saat *sprint review meeting*, dilakukan *integration testing* terhadap semua fitur yang sudah diimplementasikan pada *sprint-sprint* sebelumnya. Pada *sprint* terakhir dilakukan *system testing* untuk menguji keseluruhan fitur yang telah dikembangkan. Hasil pengujian dapat dilihat pada Tabel 5.

**Tabel 5.** Hasil Pengujian Sistem

| No | Product Backlog Item  | Prioritas | Status Uji  |
|----|---|-----------|-------------|
| 1  | Sebagai mahasiswa, saya ingin melihat jadwal kelas hari ini   | Tinggi    | OK          |
| 2  | Sebagai operator saya ingin dapat mengubah sebuah sesi kelas supaya sesuai dengan kondisi yang ada real time.       | Tinggi    | OK          |
| 3  | Sebagai mahasiswa saya ingin melihat siapa dosen yang mengajar  | Tinggi    | OK          |
| 4  | Sebagai mahasiswa saya ingin melihat informasi kelas yang batal   | Tinggi    | OK          |
| 5  | Sebagai mahasiswa saya ingin melihat ruang di mana kelas saya berada  | Tinggi    | OK          |
| 6  | Sebagai operator saya ingin dapat menginput jadwal setiap mata kuliah ke sistem.                                    | Sedang    | OK          |
| 7  | Sebagai mahasiswa, saya ingin melihat jadwal kelas hari ini dengan tampilan autoscroll                              | Sedang    | OK          |
| 8  | Sebagai operator saya ingin status sebuah jadwal kelas berubah secara otomatis berdasarkan current time.            | Rendah    | Tidak dites |
| 9  | Sebagai operator saya ingin terautentikasi untuk mengakses menu-menu operator                                       | Rendah    | Tidak dites |
| 10 | Sebagai mahasiswa saya ingin menerima pemberitahuan via email jika ada perubahan pada jadwal                        | Rendah    | Tidak dites |
| 11 | Sebagai operator saya ingin dapat mengupload jadwal kuliah dalam bentuk excel/csv agar tidak perlu menginput manual | Rendah    | Tidak dites |

Berdasarkan data hasil uji, fitur yang berhasil diimplementasikan sebanyak 7 fitur, sedangkan fitur yang tidak diimplementasikan sebanyak 4 fitur dengan prioritas rendah. Dari 7 fitur yang selesai diimplementasikan, hasil uji menunjukkan 100% atau seluruh fitur sudah sesuai dengan *requirement* sistem dan dapat digunakan dengan baik. Hal ini menggambarkan bahwa pemanfaatan Scrum dalam implementasi cukup membantu tim menyelesaikan pekerjaan-pekerjaan prioritas. Untuk fitur-fitur yang tidak diimplementasikan bukan berarti bahwa tim tidak berhasil menyelesaikan sesuai target, namun dalam pertimbangan mengenai prioritas fitur bersama dengan *product owner* dan ketersediaan waktu yang ada, sehingga diputuskan bahwa fitur akan diimplementasikan pada rilis berikutnya.

## 5 Kesimpulan

Dari hasil implementasi dan pengujian, dapat disimpulkan bahwa Laravel sebagai framework web yang digunakan cukup membantu dalam memberikan beberapa fitur yang sangat berguna bagi keseluruhan pengembangan sistem, baik dari arsitektur MVC, *template blade*, dan cara akses yang tergolong mudah. Penggunaan metode Scrum dalam pengembangan sistem informasi *display* jadwal perkuliahan membantu tim menghasilkan sistem yang memenuhi kebutuhan pengguna. Dalam waktu pengembangan yang relatif singkat hanya selama 5 minggu, sistem sudah dapat diselesaikan dapat siap untuk digunakan. Pemilihan *sprint backlog* berdasarkan prioritas serta agenda *daily scrum* yang efektif membantu tim untuk dapat menghasilkan sistem yang sesuai kebutuhan. Selain itu keterbukaan dan saling membantu antar anggota tim dan kesempatan untuk terus mempelajari hal baru juga bisa didapatkan melalui penerapan metode Scrum.

Untuk pengembangan selanjutnya, penelitian dapat dikembangkan dengan mengintegrasikan sistem *display* jadwal perkuliahan ini dengan sistem informasi akademik sehingga data jadwal dapat disinkronisasikan dengan lebih efisien. Selain itu dalam penerapan metode Scrum, dapat melibatkan *product owner* lebih banyak terutama dalam pemilihan desain dan tampilan yang diharapkan.

## Referensi

- [1] F. C. Permana, A. Kusuma, S. Prawatya, S. Sylviani, dan A. N. Wiriadikusumah, "Rancang bangun sistem informasi display jadwal perkuliahan kampus upi di cibiru berbasis web," *INTEGRATED (Information Technology and Vocational Education)*, vol. 2, no. 1, hlm. 7–14, 2020, [Daring]. Tersedia pada: <http://jadwal.kd-cibiru.upi.edu/>
- [2] A. Triyono *dkk.*, "Implementasi Raspberry Pi Untuk Aplikasi Signate Board Jadwal Kuliah Berbasis Web Dengan Sistem Operasi Linux," 2018.
- [3] I. Gede Handika dan A. Purbasari, "Konferensi Nasional Sistem Informasi 2018 STMIK Atma Luhur



- Pangkalpinang,” 2018.
- [4] Mgs. A. Firdaus, D. R. Indah, dan Idris, “Penerapan Scrum Agile Development Dalam Pengembangan Sistem Informasi Monitoring Mahasiswa Bidikmisi Berbasis Web (Studi Kasus Di Universitas Sriwijaya),” *Konferensi Nasional Teknologi Informasi dan Aplikasinya*, vol. 4, 2016.
  - [5] T. Rizaldi, D. P. S. Setyohadi, dan H. Y. Riskiawan, “Implementasi Metodologi SCRUM dalam Pengembangan Sistem Pembayaran Elektronik Pada Usaha Mikro Kecil Menengah,” *Seminar Hasil Penelitian dan Pengabdian Masyarakat Dana BOPTN Tahun 2016*, 2016.
  - [6] A. Nuraminah, “Analisis Tingkat Kematangan Manajemen Proyek Pengembangan Perangkat Lunak Menggunakan Scrum Maturity Model: Studi Kasus PT. XYZ,” *I-STATEMENT*, vol. 2, no. 2, hlm. 1–16, 2016.
  - [7] U. Ependi, “Implementasi Model Scrum pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang,” *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 3, no. 1, 2018.
  - [8] M. Billah dan A. Nuraminah, “Pengembangan Sistem E-Commerce dengan Framework Laravel dan Vuejs Menggunakan Metode Scrum Mu’tashim Billah,” *JURNAL TEKNIK INFORMATIKA DAN ELEKTRO*, vol. 4, no. 2, hlm. 130–136, 2022, [Daring]. Tersedia pada: <https://jurnal.ugp.ac.id/index.php/jutei>
  - [9] I. Ali Mohamed dan J. Yang, “Identifying the challenges of communication in implementing Agile Scrum in Embedded system development in distributed team,” 2021. [Daring]. Tersedia pada: [www.bth.se](http://www.bth.se)
  - [10] S. Sauda, N. Oktaviani, dan M. Bunyamin, “Implementasi Metode Scrum Dalam Pengembangan Test Engine Try Out Sertifikasi,” *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 3, no. 3, hlm. 70, Agu 2019, doi: 10.14421/jiska.2019.33-07.