

PENGEMBANGAN SIMIPA BERBASIS ANDROID MENGGUNAKAN METODE SCRUM PADA MODUL MAHASISWA

¹Ardiansyah, ²Dwi Sakethi, ³Anie Rose Irawati & ⁴Kelvin Putra

^{1,2,3,4}Jurusan Ilmu Komputer FMIPA Universitas Lampung, Bandar Lampung 35145
e-mail: ¹ardiansyah@fmipa.unila.ac.id, ²dwijim@fmipa.unila.ac.id, ³anie.roseirawati@fmipa.unila.ac.id,
⁴kelvinputra0210@gmail.com

Abstract — Academic activities are a learning process that is carried out based on the curriculum in university to produce quality graduate students. Good academic activities can be supported by having a system that helps academic activities at a university. FMIPA University of Lampung currently has a web-based MIPA information system (SIMIPA) which is used as a system that can support the administrative service process for FMIPA University Lampung students. Information technology that is currently developing rapidly shows that there is a need for development to be done on the SIMIPA as an effort to increase the benefits of SIMIPA. Development of an android-based student module SIMIPA application is one of the systems developed to improve services for FMIPA students at the University of Lampung. This application development will use the Scrum method. The scrum method is a software development framework from implementing Agile Development method for software projects. Development of this application is based on the product backlog which is divided into 8 sprints. The result of this research is the successful development of the SIMIPA application module for academic students using Scrum method. This is evidenced by testing using black box testing which shows the overall functional needs of the student module SIMIPA application as expected.

Keywords: Android; Application Mobile; Black box Testing; Scrum; SIMIPA

1. PENDAHULUAN

Kegiatan akademik merupakan proses pembelajaran yang dilaksanakan berdasarkan kurikulum dalam perguruan tinggi untuk mencetak lulusan mahasiswa yang berkualitas. Kegiatan akademik yang baik dapat ditunjang dengan memiliki sistem yang membantu kegiatan akademik pada suatu perguruan tinggi. FMIPA Universitas Lampung saat ini telah memiliki sistem informasi MIPA (SIMIPA) berbasis web. SIMIPA web saat ini hanya berfokus pada fitur unduh berkas ajuan layanan perkuliahan bagi mahasiswa. Formulir yang telah diajukan oleh mahasiswa kemudian hanya dapat dicek secara langsung melalui loket layanan fakultas dan tidak dapat diketahui secara *real time* untuk proses *tracking* ajuan layanan. Teknologi informasi di era ini terus mengalami pertumbuhan yang sangat pesat. Hal ini menunjukkan perlu adanya pengembangan yang harus dilakukan pada SIMIPA sebagai salah satu upaya untuk meningkatkan kegunaan dari pelayanan SIMIPA serta mengurangi kendala yang ada. Beberapa kendala yang ada antara lain mahasiswa yang sering lupa jadwal perkuliahan atau seminar, kurangnya informasi *tracking* layanan akademik dan lainnya.

Perangkat *mobile* Android saat ini telah berkembang pesat di tengah masyarakat, khususnya bagi mahasiswa. Perangkat ini memberikan kemudahan yang ditawarkan seiring perkembangannya. Kemudahan ini mendorong banyak pengembangan aplikasi pada perangkat *mobile* Android yang bertujuan memudahkan akses informasi oleh pengguna. Penggunaan perangkat *mobile* Android ini akan sangat sesuai jika dilakukan pengembangan aplikasi yang dapat digunakan mahasiswa untuk menjalankan proses kegiatan akademik.

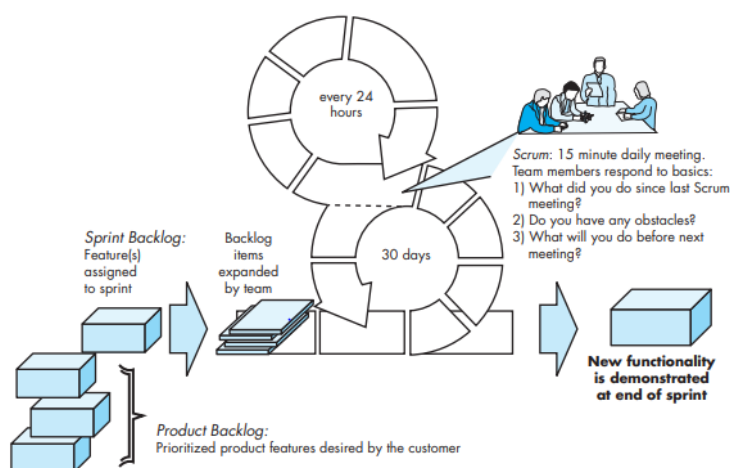
Pengembangan Sistem Informasi MIPA (SIMIPA) modul mahasiswa ini akan dikembangkan dengan menggunakan metode *scrum*. *Scrum* merupakan *framework* yang digunakan untuk implementasi metode pengembangan aplikasi *agile*. *Agile* ialah metodologi pengembangan aplikasi yang memerlukan proses adaptasi cepat terhadap segala bentuk perubahan [1]. Pemilihan metode *scrum* dalam penelitian ini memperhatikan terkait pengembangan SIMIPA yang akan dibagi ke dalam beberapa tim, yaitu SIMIPA modul

mahasiswa, SIMIPA modul dosen dan SIMIPA berbasis Web. Penelitian yang terdiri dari tim-tim tersebut menjadi faktor utama pemilihan *scrum* karena metode *scrum* sangat cocok digunakan secara tim agar proses pengembangan perangkat lunak dapat saling terkoordinasi dan terstruktur antara anggota tim yang memiliki berbagai tugas yang berbeda [2].

Berdasarkan uraian latar belakang tersebut, hasil yang ingin dicapai adalah mengembangkan SIMIPA berbasis Android menggunakan metode *scrum* pada modul mahasiswa.

2. METODOLOGI PENELITIAN

Pengembangan SIMIPA berbasis Android pada modul mahasiswa dikembangkan menggunakan metode pengembangan *scrum*. Metode *scrum* merupakan sebuah metode kerangka kerja untuk menyelesaikan permasalahan yang kompleks, adaptif untuk menghasilkan produk yang bernilai tinggi dengan kreativitas serta produktivitas kerja pada waktu yang bersamaan [3]. Pengembangan aplikasi menggunakan *scrum* akan dikerjakan secara tim yaitu oleh *scrum team*. *Scrum team* terdiri dari *product owner*, *scrum master* dan tim pengembang. *Product owner* adalah pihak yang menentukan fungsi apa yang akan dikembangkan selama penelitian berlangsung, *scrum master* sebagai pihak yang memandu tim *scrum* bekerja agar penelitian dapat berjalan sesuai dengan kaidah *scrum*, serta tim pengembang yang bertugas untuk mengembangkan produk [4]. Tahapan pengembangan aplikasi menggunakan *scrum* dapat dilihat pada Gambar 1 [5].



Gambar 1. Tahapan Metode *scrum*

2.3.1 Product Backlog

Tahap awal dari metode *scrum* adalah penyusunan *product backlog*. *Product backlog* adalah daftar kebutuhan atau fitur yang memberikan nilai bisnis klien, dan produk backlog yang dibuat dapat bertambah [6]. *Product backlog* dalam penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Product Backlog SIMIPA Modul Mahasiswa

No	Product Backlog	Penanggung Jawab
1	Use case diagram mahasiswa	Kelvin
2	Sequence diagram mahasiswa	Kelvin
3	Activity diagram mahasiswa	Kelvin
4	Perancangan UI home	Kelvin
5	Perancangan UI intro dan <i>Splash screen</i>	Kelvin
6	Perancangan UI Login	Kelvin
7	Perancangan UI presensi kehadiran seminar	Kelvin
8	Perancangan UI Kalender Akademik	Kelvin
9	Perancangan UI Informasi Layanan	Kelvin
10	Perancangan UI Berita FMIPA	Kelvin
11	Perancangan UI Tambah Prestasi Mahasiswa	Kelvin
12	Implementasi <i>mockup</i> ke xml	Kelvin
13	Pengembangan Fungsi <i>Home</i>	Kelvin
14	Pengembangan Fungsi Intro dan <i>Splash screen</i>	Kelvin
15	Pengembangan Fungsi Login	Kelvin
16	Pengembangan Fungsi Kehadiran Seminar	Kelvin
17	Pengembangan Fungsi Kalender Akademik	Kelvin
18	Pengembangan Fungsi Informasi Layanan	Kelvin
19	Pengembangan Fungsi Berita FMIPA	Kelvin
20	Pengembangan Fungsi Tambah Prestasi Mahasiswa	Kelvin

2.3.2 Sprint Planning

Sprint planning dilakukan oleh *product owner* untuk menjelaskan *product backlog* kepada *development team*. *Product backlog* pada *sprint planning* akan dibagi lagi menjadi *sprint goal* dan *sprint backlog* [7].

2.3.3 Daily Scrum

Daily scrum dilakukan oleh seluruh tim SIMIPA untuk mengerjakan *sprint backlog* masing-masing. Tahap ini memungkinkan anggota tim pengembangan untuk membagikan pemikiran mereka dalam bentuk saran serta mendiskusikan semua hambatan yang mereka miliki selama *daily scrum* [8].

2.3.4 Sprint Review

Sprint review dilakukan oleh seluruh tim untuk mempresentasikan hasil pekerjaan yang telah dilakukan. Tujuan dari *sprint review* adalah untuk memeriksa dan mengadaptasi produk yang sedang dibangun [9]. *Sprint* yang telah dikerjakan kemudian akan dilakukan *review* serta dilakukan pengujian dengan menggunakan metode *black box testing*. *Black box testing* merupakan teknik pengujian perangkat lunak yang menguji *requirements* dan fungsional dari perangkat lunak [10]. Teknik *black box* yang akan digunakan adalah *equivalence partitions*. *Equivalence partitions* adalah metode pengujian yang memecah atau membagi domain masukan dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh [11].

2.3.5 Sprint Retrospective

Tahap ini dilakukan untuk pemberian masukan dari *scrum master* kepada *development team*. Kemudian *scrum team* membahas tugas mana yang mengalami masalah, tugas mana yang bisa dilanjutkan dan yang tidak bisa dilanjutkan ke *sprint* berikutnya [12].

3. HASIL DAN PEMBAHASAN

Pengembangan aplikasi ini berdasarkan *product backlog* yang telah disusun sebelumnya oleh *product owner*. *Product backlog* kemudian akan dibagi ke dalam delapan *sprint* yang akan dikerjakan. Pengembangan aplikasi ini akan terjadi pertukaran data dari server dengan aplikasi *mobile* menggunakan API yang dikerjakan oleh anggota tim lain. API merupakan antarmuka yang menghubungkan antara komputer dan server untuk mengakses data dan fungsi [13]. Format pertukaran data yang digunakan pada API dalam penelitian ini adalah *Javascript Object Notation (JSON)*. JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia serta mudah diterjemahkan oleh komputer yang dibuat berdasarkan bahasa pemrograman javascript [14]. Pertukaran data menggunakan API melalui perintah HTTP yang bisa digunakan dalam REST adalah GET, POST, PUT dan DELETE [15].

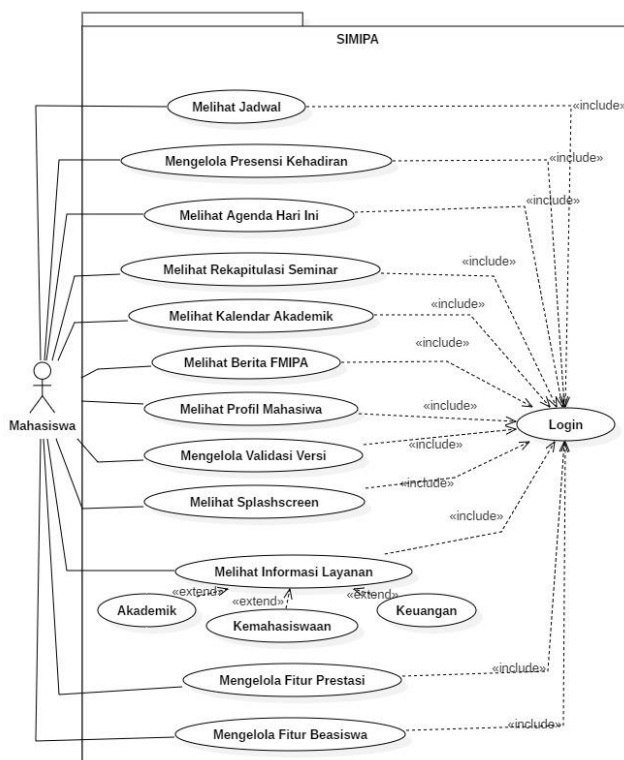
3.1. Sprint ke-1

3.1.1 Sprint Planning

Sprint planning ke-1 menghasilkan tiga *sprint backlog* yang akan dikerjakan. *Sprint backlog* yang akan dikerjakan terdiri dari pembuatan *use case diagram*, *activity diagram* dan *sequence diagram*.

3.1.2 Daily Scrum

Daily scrum pada *sprint* ke-1 menghasilkan diagram-diagram *activity*, *sequence* dan *use case*. Pembuatan diagram ini menggunakan *software Star UML*. Diagram-diagram ini berguna sebagai bentuk rancangan proses bisnis pada aplikasi SIMIPA. Hasil *use case diagram* dapat dilihat pada Gambar 2.



Gambar 2. Use Case Diagram SIMIPA.

3.1.3 *Sprint Review*

Anggota *development team* akan mempresentasikan hasil proses pengerjaan masing-masing pada *sprint* ke-1 yang kemudian akan diberi *feedback* oleh *scrum master* berupa saran terkait hasil *sprint*.

3.1.4 *Sprint Retrospective*

Sprint retrospective membahas kendala, saran dan solusi pada *sprint* ke-1. *Retrospective* pada *sprint* pertama adalah melakukan perbaikan dengan penambahan relasi di setiap *use case* dengan relasi *include/extend*.

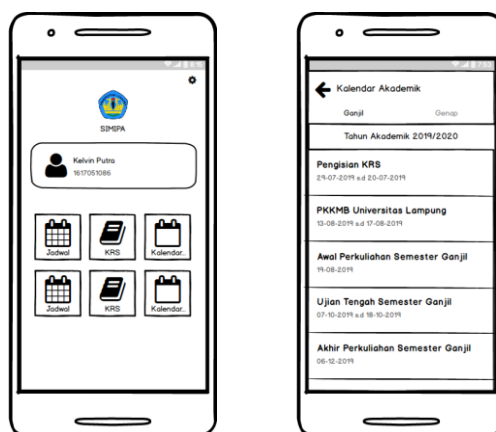
3.2. *Sprint ke-2*

3.2.1 *Sprint Planning*

Sprint planning pada *sprint* ke-2 menghasilkan delapan *sprint backlog* yang harus dikerjakan pada *sprint* ini yaitu, perancangan UI *home*, perancangan UI *intro/splash screen*, perancangan UI *login*, perancangan UI kehadiran seminar, perancangan UI kalender akademik, perancangan UI informasi layanan, perancangan UI berita FMIPA dan perancangan UI tambah prestasi.

3.2.2 *Daily Scrum*

Hasil pelaksanaan *daily scrum* pada *sprint* ke-2 berupa rancangan *mockup* yang dibuat menggunakan *software Balsamiq Mockup*. Contoh hasil rancangan *mockup* dapat dilihat pada Gambar 3.



Gambar 3. Tampilan Rancangan *Mockup* Aplikasi SIMIPA.

3.2.3 *Sprint Review*

Sprint review pada *sprint* ke-2 diikuti oleh seluruh anggota *scrum team*. Hasil pengerjaan *mockup* pada *sprint* ke-2 kemudian dilakukan presentasi oleh *development team* kepada *scrum master*.

3.2.4 *Sprint Retrospective*

Hasil *retrospective* pada *sprint* ke-2 yaitu terdapat perbaikan yang harus dilakukan sebelum melanjutkan ke *sprint* ke-3. Perbaikan yang harus dilakukan yaitu pada fitur informasi layanan akan berupa *history* ajuan layanan mahasiswa.

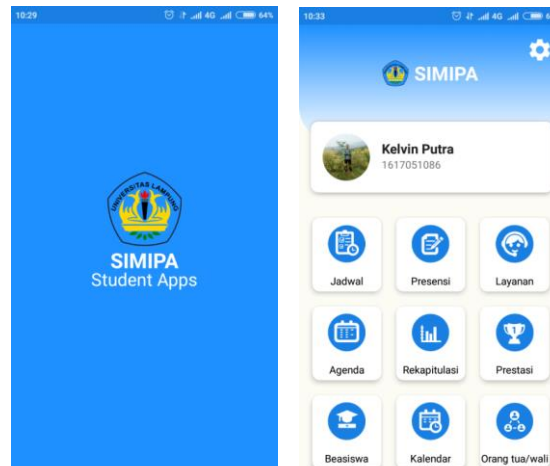
3.3. *Sprint ke-3*

3.3.1 *Sprint Planning*

Sprint planning ke-3 terdapat 1 *sprint backlog* yang akan dikerjakan dalam pembuatan aplikasi SIMIPA modul mahasiswa yaitu implementasi *mockup* ke dalam xml dengan tingkat kesulitan tinggi.

3.3.2 Daily Scrum

Daily scrum dalam *sprint* ini melakukan pekerjaan pembuatan seluruh *user interface* menggunakan xml di Android Studio berdasarkan rancangan *mockup* yang telah dikerjakan pada *sprint* ke-2. Hasil implementasi *mockup* ke dalam xml dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Hasil Implementasi *Mockup* ke XML.

3.3.3 Sprint Review

Sprint review dilaksanakan oleh seluruh anggota *scrum team* untuk membahas pengerjaan pada *sprint* ke-3. Anggota *development team* kemudian masing-masing mempresentasikan hasil pengerjaan *sprint* pada *sprint* ke-3.

3.3.4 Sprint Retrospective

Scrum master memberikan saran yang kemudian menjadi *retrospective* untuk anggota *development team*. *Retrospective* ini harus diselesaikan sebelum melanjutkan ke *sprint* ke-4. *Retrospective* pada *sprint* ke-3 yaitu menambahkan nama SIMIPA pada menu *home*, menyeragamkan ukuran *text size* pada nama menu dan warna biru pada aplikasi disamakan dengan warna biru FMIPA.

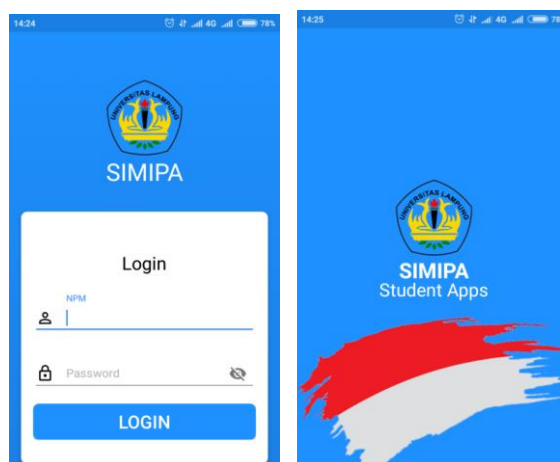
3.4. Sprint ke-4

3.4.1 Sprint Planning

Sprint planning dilakukan oleh seluruh *scrum team* untuk membahas perencanaan pengerjaan pada *sprint* ke-4. Hasil *sprint planning* ini menghasilkan pembagian *product backlog* menjadi 3 *sprint backlog* untuk pengembangan aplikasi SIMIPA modul mahasiswa yaitu pengembangan fungsi *intro app* dan *Splash screen*, pengembangan fungsi *login* dan pengembangan fungsi *home*.

3.4.2 Daily Scrum

Daily scrum pada *sprint* ini telah berfokus pada pengembangan fungsi untuk fitur SIMIPA modul mahasiswa. Pengembangan fungsi ini menggunakan bahasa pemrograman Java. Fungsi yang dikembangkan dalam *sprint* ini yaitu fungsi *intro app/splash screen*, pengembangan fungsi *login* berisi fungsi untuk *user* bisa masuk ke dalam aplikasi dan fitur *home* berisi profil mahasiswa dan menu-menu yang terdapat pada aplikasi SIMIPA modul mahasiswa. Hasil pengembangan fungsi pada *sprint* ini dapat dilihat pada Gambar 5.



Gambar 5. Tampilan Hasil Pengembangan Fungsi *Sprint* ke-4.

3.4.3 *Sprint Review*

Sprint review dilaksanakan oleh seluruh *scrum team* untuk melaksanakan presentasi hasil pengerjaan pada *sprint* ini. Tahap ini dilakukan juga pengujian terhadap pengembangan fungsi-fungsi pada *sprint* ke-4 dengan menggunakan metode *black box testing*. Hasil pengujian pada fitur ini valid seluruhnya.

3.4.4 *Sprint Retrospective*

Scrum master dan *development team* secara keseluruhan telah menyetujui hasil pengerjaan pada *sprint* ke-4 dan langsung dapat melanjutkan ke *sprint* ke-5.

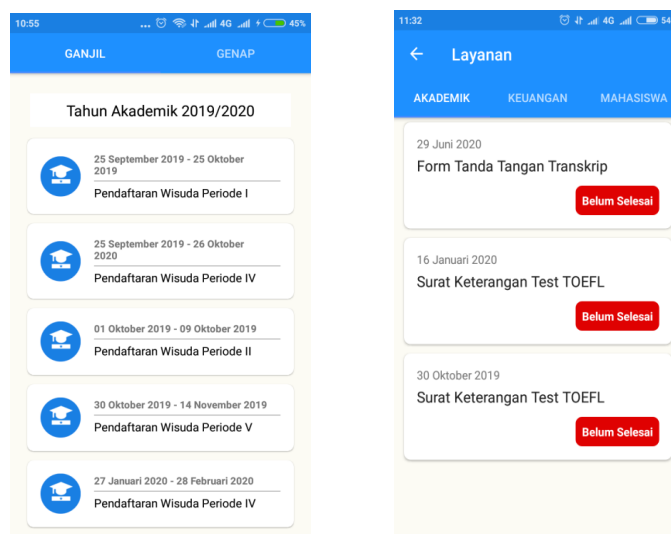
3.5. *Sprint* ke-5

3.5.1 *Sprint Planning*

Sprint backlog yang akan dikerjakan pada *sprint* ke-5 adalah pengembangan fungsi kalender akademik dan pengembangan fungsi informasi layanan dengan memiliki tingkat kesulitan tinggi.

3.5.2 *Daily Scrum*

Pengembangan fungsi yang dilakukan dalam *daily scrum* ini adalah pengembangan fungsi untuk fitur kalender akademik dan informasi layanan. Hasil pengerjaan *daily scrum sprint* ke-5 dapat dilihat pada Gambar 6.



Gambar 6. Tampilan Hasil Pengembangan Fungsi *Sprint* ke-5.

3.5.3 *Sprint Review*

Sprint review kembali dilaksanakan dengan mempresentasikan hasil pengerjaan *sprint*. Pengujian menggunakan *black box testing* dilakukan guna menguji fungsional pada fitur kalender akademik dan informasi layanan yang telah dikerjakan. Hasil pengujian pada *sprint review* ini menunjukkan seluruh fungsional berhasil dikembangkan sesuai kelas uji masing-masing.

3.5.4 *Sprint Retrospective*

Retrospective pada *sprint* ke-5 terdapat perbaikan yaitu pada tampilan di *fragment* semester genap di menu kalender akademik. Perbaikan ini harus dilaksanakan sebelum melanjutkan ke *sprint* berikutnya.

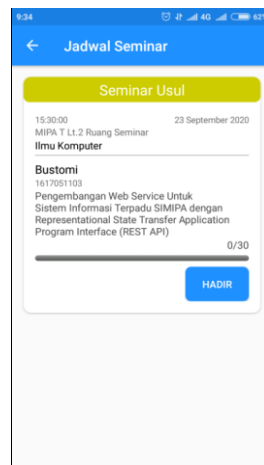
3.6. *Sprint* ke-6

3.6.1 *Sprint Planning*

Sprint planning ke-6 dilakukan oleh seluruh anggota *scrum team*. Pertemuan ini menghasilkan satu *sprint backlog* yang akan dikerjakan yaitu pengembangan fungsi kehadiran seminar dengan tingkat kesulitan tinggi,

3.6.2 *Daily Scrum*

Pengerjaan *daily scrum* pada *sprint* ke-6 mengerjakan pengembangan fungsi kehadiran seminar. Pengembangan fungsi ini meliputi fungsi menampilkan *list* jadwal seminar mahasiswa FMIPA, kemudian fungsi untuk mahasiswa dapat melakukan konfirmasi kehadiran seminar atau batal hadir ke seminar. Hasil pengembangan fungsi ini dapat dilihat pada Gambar 7.



Gambar 7. Tampilan Hasil Pengembangan Fungsi Sprint ke-6.

3.6.3 *Sprint Review*

Sprint review ini dilakukan pengujian meliputi fungsi *list* jadwal seminar dan konfirmasi kehadiran atau batal pada seminar yang tersedia. Hasil pengujian fitur ini yaitu valid untuk keseluruhan fungsional.

3.6.4 *Sprint Retrospective*

Sprint retrospective pada *sprint* ke-6 terdapat 2 perbaikan yaitu jadwal seminar yang telah berlalu tidak perlu ditampilkan dan pemberian nama jurusan pada jadwal seminar. Perbaikan ini dikerjakan sebelum melanjutkan ke *sprint* ke-7.

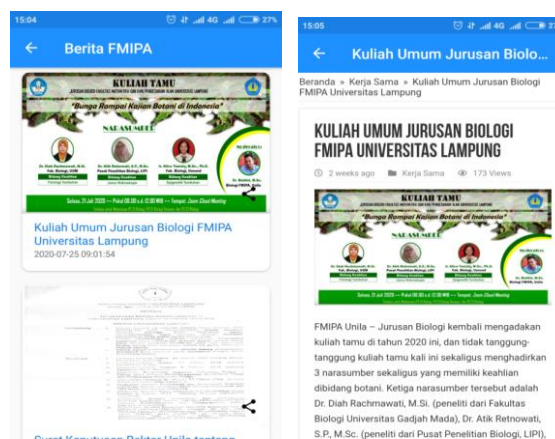
3.7. *Sprint* ke-7

3.7.1 *Sprint Planning*

Sprint backlog yang dihasilkan pada *sprint planning* yaitu pengembangan fungsi fitur berita FMIPA dengan tingkat kesulitan tinggi.

3.7.2 *Daily Scrum*

Pengerjaan *daily scrum* pada *sprint* ke-7 yaitu mengembangkan fungsi pada fitur berita FMIPA meliputi menampilkan daftar berita serta detail berita FMIPA. Fitur berita FMIPA pada aplikasi ini terintegrasi dengan berita di website fmipa.unila.ac.id. Hasil pengembangan fungsi ini dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Hasil Pengembangan Fungsi Sprint ke-7.

3.7.3 *Sprint Review*

Pengujian pada *sprint* ini dilakukan dengan menguji pada fitur daftar dan detail berita FMIPA dengan menggunakan *black box testing*. Hasil pengujian ini menyatakan valid untuk keseluruhan fungsional.

3.7.4 *Sprint Retrospective*

Kendala yang timbul dalam *sprint* ini adalah data *response* untuk detail berita mengandung *tag* HTML sehingga tampilan dalam detail berita berantakan. Solusi untuk permasalahan ini menghasilkan solusi untuk membuat *activity* detail berita FMIPA berupa *webview*.

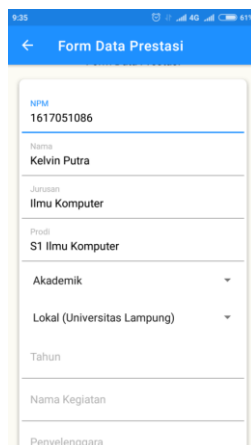
3.8. *Sprint ke-8*

3.8.1 *Sprint Planning*

Sprint planning ke-8 menghasilkan 1 *sprint backlog* yang akan dikerjakan yaitu pengembangan fungsi tambah data prestasi mahasiswa.

3.8.2 *Daily Scrum*

Pada *daily scrum* ke-8 dilakukan penambahan fungsi untuk fitur tambah data prestasi mahasiswa FMIPA Universitas Lampung. Mahasiswa dapat menambahkan data prestasi masing-masing melalui fitur ini. Hasil pengerjaan *daily scrum* pada *sprint* ke-8 dapat dilihat pada Gambar 9.



Gambar 9. Tampilan Hasil Pengembangan Fungsi Sprint ke-8.

3.8.3 *Sprint Review*

Pengujian fungsional saat *sprint* ke-8 kemudian dilakukan untuk menguji fitur pada pengembangan fungsi tambah data prestasi dengan hasil pengujian valid untuk fungsional yang telah dikembangkan.

3.8.4 *Sprint Retrospective*

Sprint ke-8 dalam pengerjaannya terdapat kendala dalam pengembangan untuk *upload* foto. Kendala ini mengakibatkan waktu pengerjaan mengalami penambahan waktu selama 2 minggu.

4. KESIMPULAN

Pengembangan aplikasi SIMIPA modul mahasiswa bidang akademik dengan menggunakan metode *scrum* berhasil dilaksanakan sesuai dengan *product backlog* yang telah dibuat berjumlah 20 dan dibagi ke dalam 8 *sprint* dalam total waktu pengerjaan selama 23 minggu.

Hasil pengujian yang menggunakan metode *black box testing* pada *sprint* ke-4, *sprint* ke-5, *sprint* ke-6, *sprint* ke-7 dan *sprint* ke-8 menunjukkan keseluruhan fungsional pada aplikasi SIMIPA modul mahasiswa dapat berjalan dengan baik dan telah disetujui oleh *product owner* dan *scrum team*.

DAFTAR PUSTAKA

- [1] C. E. Prastio and N. Ani, "Aplikasi Self Service Menu Menggunakan Metode Scrum Berbasis Android (Case Study: Warkobar Cafe Cikarang)," *JURNAL PETIR*, vol. 11, no. 2, pp. 203-220, 2018.
- [2] F. M. Prabowo, Trisyanto, Ardiansyah and A. Hijriani, "Pengembangan Aplikasi Marketplace Berbasis Android Dengan Metode Scrum (Studi Kasus: Marketplace Pakan Ternak dan Produk Peternak Epakan.id Modul Penjual)," in *SNSMIAP 2019*, Bandar Lampung, 2019.
- [3] K. Schwaber and J. Sutherland, "Scrum Guide," November 2017. [Online]. Available: <https://scrumguides.org/>. [Accessed 2021].
- [4] F. Maulana and D. A. Ramadhan, "Pengembangan Aplikasi Android Patriot Pangan sebagai Sarana e-Participation untuk Sistem Ketahanan Pangan Nasional," *Jurnal Ilmu Komputer Dan Agri-Informatika*, vol. 7, no. 2, pp. 124-134, 2020.
- [5] R. S. Pressman and B. R. Maxim, *Software engineering : a practitioner's approach Eighth edition*, New York: McGraw-Hill Education, 2015.
- [6] M. A. Dewi and R. Andriyani, "Implementasi Scrum Model Development Pada Monitoring Inventory Control Cleaning Equipment dan Chemical PT. Explore Global Solution," *Jurnal ULTIMA InfoSys*, vol. 8, no. 2, pp. 112-117, 2018.
- [7] J. Arka, A. H. Brata and K. C. Brata, "Pengembangan Aplikasi Mobile Manajemen Keuangan Dengan Metode Scrum (Studi Kasus Mahasiswa FILKOM UB)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 2, pp. 1199-1207, 2018.
- [8] K. Bhavsar, S. Gopalan and V. Shah, "Scrum: An Agile Process Reengineering in Software Engineering," in *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Ahmedabad, 2020.
- [9] B. Septian, I. K. Jayadi, M. Holil and I. Handriani, "Sistem Human Capital Management Menggunakan Metode Scrum," *Jurnal Sistem Informasi, Teknologi Informatika dan Komputer*, vol. 11, no. 1, pp. 1-16, 2020.
- [10] S. R. Jan, S. T. Ullah Shah, Z. U. Johar and Y. Shah, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *IJRSET*, vol. 2, no. 2, p. 682-689, 2016.
- [11] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. Ardhie Mardangga, M. Astuti and A. Saifudin, "Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 2, no. 4, pp. 155-161, 2019.
- [12] P. A. Guna Permana, "Scrum Method Implementation in a Software Development Project Management," *IJACSA*, vol. 6, no. 9, pp. 198-204, 2015.
- [13] R. A. Syach and D. A. Ramadhan, "Pengembangan Aplikasi Android E-Initiative Patriot Pangan sebagai Platform Crowdfunding Kebutuhan Balita Menggunakan REST API," *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 7, no. 2, pp. 84-95, 2020.
- [14] A. Aziz, W. and B. Wicaksono, "Pemanfaatan Web Service Moodle Berbasis REST-JSON untuk Membangun Moodle Online Learning Extension berbasis Android," *Jurnal ITSMART*, vol. 2, no. 2, pp. 1-6, 2013.

- [15] M. A. Rahman, I. Kuswardayan and R. R. Hariadi, "Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android," *Jurnal Teknik POMITS*, vol. 2, no. 1, pp. 1-4, 2013.