

Sistem Informasi *Monitoring* Perkuliahan Terintegrasi Universitas Sam Ratulangi

Geraldo F.T.J Lantang¹⁾, Sherwin R U Sompie²⁾, Alwin Melkie Sambul³⁾

Teknik Elektro Universitas Sam Ratulangi, Jl. Kampus Bahu-Unsrat Manado, Indonesia 95115

E-mail: 15021106163@unsrat.ac.id¹⁾, aldosomepie@unsrat.ac.id²⁾, asambul@unsrat.ac.id³⁾

Diterima:tgl;direvisi:tgl;disetujui:tgl

Abstract — *The implementation of Information Systems in an organization has a role in processing activities or transactions in which there are management staff, media, procedures, facilities and technology to produce informative data that can later be integrated with other parties. In the current era, the development of Information Systems has become crucial that requires almost all service agencies to manage their information in an integrated manner. Related to Academy Institutions, the amount of information concerning all components in the university requires accuracy in processing the relevant data. Through the use of information systems, various information can be accessed by various universities to facilitate the management of university activities. Seeing the problems that occur, conducted a study held to help the efficiency of data management and monitoring of lecture activities, by helping to digitize some manual processes.*

Keywords : CodeIgniter, Information System, Integrated, RFID, Scrum Method, University.

Abstrak— Penerapan Sistem Informasi dalam suatu organisasi berperan dalam memproses aktivitas atau transaksi dimana di dalamnya ada tenaga pengelola, media, prosedur, fasilitas serta teknologi untuk menghasilkan data informatif yang nantinya dapat terintegrasi dengan pihak lain. Di era kini, pengembangan Sistem Informasi telah menjadi hal krusial yang menuntut hampir seluruh badan layanan atau instansi untuk melakukan pengelolaan informasi masing-masing secara terintegrasi. Berkaitan dengan Instansi Perguruan Tinggi, banyaknya informasi yang menyangkut mengenai segala komponen dalam universitas mengharuskan ketepatan dalam pengolahan data yang bersangkutan.

Kata Kunci: CodeIgniter, Integrasi, RFID, Scrum Method, Sistem Informasi, Universitas.

I. PENDAHULUAN

Secara prinsip, monitoring dilakukan sementara kegiatan sedang berlangsung dengan tujuan memastikan kesesuaian proses dan capaian sesuai rencana. Bila ditemukan penyimpangan atau kelambanan maka segera dibenahi sehingga kegiatan dapat berjalan sesuai rencana dan targetnya. Jadi, hasil monitoring dibagi menjadi input bagi kepentingan proses selanjutnya, dalam hal ini monitoring dilakukan untuk meningkatkan kegiatan belajar mengajar berdasarkan data yang didapatkan pada monitoring untuk meningkatkan kualitas pembelajaran di Universitas Sam Ratulangi Manado.

Universitas Sam Ratulangi merupakan salah satu universitas unggulan yang terakreditasi A dari BAN-PTN yang terletak di kawasan Indonesia Timur tepatnya di kota Manado

ibu kota dari provinsi Sulawesi Utara . Terdiri dari 11 fakultas serta 1 pascasarjana yang memiliki 23,107 mahasiswa aktif berdasarkan tahun ajaran 2019/2020. Banyaknya jumlah mahasiswa aktif yang berkuliah di Univesitas Sam Ratulangi tentunya dibutuhkan sebuah monitoring yang dapat merekap jumlah mahasiswa yang hadir sehingga dapat membantu dalam keputusan akademik maupun peningkatan mutu pembelajaran menjadi yang lebih baik..[1]

Adapun kinerja monitoring dan rekapitulasi absensi perkuliahan yang berjalan di Universitas Sam Ratulangi tersebut dinilai masih belum optimal karena pada pengolahan data absensi mahasiswa di kampus Universitas Sam Ratulangi masih dilakukan menggunakan kertas sebagai media dalam pengisian daftar hadir. Pengolahan data jadi kurang efektif dan efisien karena kemungkinan terjadinya manipulasi atau kehilangan data yang beresiko tinggi dan bisa berdampak pada proses perkuliahan di Universitas Sam Ratulangi Manado.

Melihat masalah tersebut, penulis mengangkat topik penelitian yang bertujuan menghasilkan sebuah Sistem Informasi Monitoring Perkuliahan berbasis web dengan menggunakan alat berupa kartu yang berisikan chip data identitas mahasiswa berupa Nama atau NIM.[1].

A. Sistem Informasi Monitoring

Sistem Informasi Monitoring adalah merupakan komponen yang saling bekerja sama untuk mengumpulkan, mengolah, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengendalian, analisis masalah dan visualisasi dalam sebuah organisasi. Selain itu, (Stair & Reynolds, 2012) mendefinisikan sistem informasi sebagai seperangkat elemen atau komponen yang saling terkait yang dikumpulkan (*input*), manipulasi (*process*), menyimpan, dan menyebarkan (*output*) data dan informasi dan memberikan reaksi korektif (*feedback*) untuk memenuhi tujuan. [2] Menurut *Wikipedia monitoring* adalah pemantauan yang dapat dijelaskan sebagai kesadaran (*awareness*) tentang apa yang ingin diketahui, pemantauan berkadar tingkat tinggi dilakukan agar dapat membuat pengukuran melalui waktu yang menunjukkan pergerakan ke arah tujuan atau menjauh dari itu. Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan [2].

masih bernama *Form Interpreted* (FI), yang berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.[5]

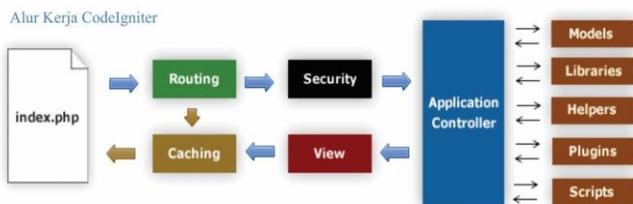
B. Aplikasi Berbasis Web

Aplikasi seperti ini pertama kali dibangun hanya dengan menggunakan bahasa yang disebut *HTML* (*HyperText Markup Language*) dan protokol yang digunakan dinamakan *HTTP* (*HyperText Transfer Protocol*). Semua perubahan harus dilakukan pada level aplikasi. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan *HTML*.

Konsep yang mendasari aplikasi Web sebenarnya sederhana. Operasi yang melatarbelakanginya melibatkan pertukaran informasi antara komputer yang meminta informasi, yang disebut klien, dan komputer yang memasok informasi (atau disebut server). [3] Secara lebih detail, server yang melayani permintaan dari klien yang sesungguhnya berupa suatu perangkat lunak yang dinamakan Web server. [3]

C. CodeIgniter

Sebuah *web application network* yang bersifat open source yang digunakan untuk membangun aplikasi *php* dinamis. Pertama kali dikembangkan pada tahun 2006 oleh Rick Ellis.[4] CodeIgniter menjadi sebuah framework PHP dengan model *MVC* (*Model, View, Controller*) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang lengkap disertai dengan contoh implementasi kodenya. Dokumentasi yang lengkap inilah yang menjadi salah satu alasan kuat mengapa banyak orang memilih CodeIgniter sebagai *framework* pilihannya[4].



Gambar 1. Alur kerja *codeigniter*



Gambar 2. *Hypertext Preprocessor*

Bahasa pemrograman yang sering disisipkan ke dalam HTML. PHP sendiri berasal dari kata Hypertext Preprocessor. Sejarah PHP pada awalnya merupakan kependekan dari *Personal Home Page* (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP

Bahasa pemrograman ini menggunakan sistem server-side. Server-side programming adalah jenis bahasa pemrograman yang nantinya script/program tersebut akan dijalankan/diproses oleh server. Kelebihannya adalah mudah digunakan, sederhana, dan mudah untuk dimengerti dan dipelajari serta sudah menjadi standar bagi programmer web di seluruh dunia. PHP juga menjadi dasar dari aplikasi CMS (*Content Management System*) populer seperti Joomla, Drupal, dan WordPress.[5]

D. Javascript

JavaScript adalah salah satu bahasa pemrograman yang digunakan untuk pengembangan sistem berbasis web dan berguna membuat website lebih interaktif. Bahasa ini sudah ada dan digunakan dalam kurun waktu dua puluh tahun ini.[6]

JavaScript dikenal oleh para pengembang website sebagai salah satu bahasa pemrograman utama bagi web developer. Dalam pengembangan website sekarang ini, javascript sudah menjadi Bahasa Pemrograman yang sangat dibutuhkan misalkan untuk membuat modul alert, notofocation, form validation, efek, ajax, game dan sebagainya.[6]

E. MariaDb

MariaDB adalah sebuah database manajemen sistem yang di kembangkan oleh pengembang MySQL, sebab MySQL sendiri telah di ambil alih oleh perusahaan ORACLE yang juga merupakan perusahaan yang bergerak dalam database sehingga menyebabkan MySQL menjadi produk yang berlisensi dan akan menjadi produk yang komersil.[7]



Gambar 3. *MariaDb*



Gambar 4. *Gitlab*

G. Gitlab

Gitlab merupakan software dikembangkan oleh Dmitriy Zaporozhets dan Valery Sizov dari Ukraina. Penulisan kode dalam Gitlab ditulis menggunakan Ruby. Dengan munculnya

Gitlab membuat para pengembang melirik dan mulai mencoba fitur-fitur yang ditawarkan oleh Gitlab.[8]

Dengan menggunakan Git, setiap pengembang yang berkolaborasi dapat melakukan perubahan pada source-code tanpa harus takut terjadi bentrok ataupun kesulitan dalam menggabungkan hasil perubahan yang mereka lakukan.[8]

F. *Sqlyog*

Sqlyog merupakan aplikasi client *MySQL* yang berguna untuk pengelolaan database dalam jumlah yang besar dan memiliki banyak table, *Sqlyog* juga digunakan sebagai remote untuk mengakses database kita yang berada di server.[9]

G. *Scrum Method*

Scrum adalah kerangka kerja proses yang telah digunakan untuk mengelola pengembangan produk kompleks sejak awal tahun 1990-an. Metode ini sangat cocok digunakan dalam pengembangan aplikasi yang kompleks. *Scrum* merupakan kerangka kerja dimana anda dapat menggunakan bermacam proses dan teknik di dalamnya. *Scrum* mengekspos ketidak-efektifan dari manajemen produk dan teknik kerja anda, sehingga anda dapat secara terus-menerus meningkatkan kinerja produk, tim, dan lingkungan kerja..[10]

1) *Scrum Team*

Scrum Team terdiri dari *Product Owner*, *Development Team* dan *Scrum Master*. *Scrum Team* bersifat swakelola dan lintas-fungsi. Tim yang swakelola memilih cara terbaik dalam mengerjakan pekerjaan mereka, bukan diperintah oleh orang lain di luar tim ini. Tim yang lintas-fungsi memiliki semua keahlian yang dibutuhkan untuk menyelesaikan pekerjaan mereka tanpa bergantung pada orang lain di luar tim ini. Bentuk tim dalam *Scrum* dirancang untuk mengoptimalkan fleksibilitas, kreativitas dan produktivitas. Bentuk *Scrum Team* telah terbukti menjadikan tim semakin efektif dalam mengerjakan semua tipe pekerjaan yang telah disebutkan sebelumnya dan untuk jenis pekerjaan kompleks apa pun. *Scrum Team* menghantarkan produk secara iteratif dan inkremental guna memaksimalkan peluang untuk mendapatkan umpan balik. Pengantaran produk “Selesai” secara inkremental dilakukan guna memastikan versi produk yang berpotensi untuk digunakan selalu siap tersedia. *Product Owner* adalah orang yang bertanggung jawab untuk memaksimalkan nilai bisnis dari produk yang dihasilkan oleh *Development Team*. [11]

2) *Sprint*

Inti dari *Scrum* adalah *Sprint*, yaitu sebuah batasan waktu dengan durasi satu bulan atau kurang, dimana terdapat proses pembuatan Increment yang “Selesai”, dapat digunakan dan berpotensi untuk dirilis. *Sprint* memiliki durasi yang konsisten sepanjang daur hidup pengembangan produk. *Sprint* yang baru langsung dimulai setelah *Sprint* sebelumnya selesai. Setiap *Sprint* bisa dianggap sebagai sebuah proyek dengan durasi tidak lebih dari satu bulan yang dijalankan untuk mencapai sebuah tujuan. Setiap *Sprint* memiliki tujuan mengenai apa yang harus dibangun, sebuah rancangan dan perencanaan fleksibel yang memandu pembangunan tersebut, daftar pekerjaan, dan hasil dari Increment. *Sprint* dibatasi tidak

lebih dari satu bulan kalender. Ketika durasi *Sprint* terlalu lama maka definisi dari apa yang akan dikembangkan dapat berubah, kompleksitas dapat meningkat, dan risiko juga dapat meningkat. Dengan memastikan adanya inspeksi dan adaptasi, keberadaan *Sprint* menciptakan sebuah tingkat kemungkinan perkembangan pekerjaan menuju *Sprint Goal* setidaknya setiap satu bulan kalender. Dengan adanya *Sprint*, risiko pengeluaran biaya dibatasi menjadi maksimal satu bulan kalender..[12]

3) *Sprint Planning*

Pekerjaan yang akan dikerjakan di *Sprint* direncanakan pada saat *Sprint Planning*. Perencanaan ini dilakukan secara kolaboratif oleh seluruh anggota *Scrum Team*. *Sprint Planning* memiliki batasan waktu maksimal delapan jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih singkat, acara ini biasanya lebih singkat. *Scrum Master* memastikan acara ini diselenggarakan dan peserta memahami tujuannya. *Scrum Master* mengedukasi *Scrum Team* untuk menjaganya di dalam batasan waktu. *Development Team* memprakirakan fungsionalitas yang bisa dikembangkan selama *Sprint*. *Product Owner* membahas obyektif yang harus dicapai di *Sprint* dan *Product Backlog* item yang dapat mencapai *Sprint Goal* bila diselesaikan.

Scrum Team berkolaborasi untuk memahami seluruh pekerjaan untuk *Sprint*. Masukan untuk pertemuan ini adalah *Product Backlog*, Increment terkini, proyeksi kapasitas *Development Team* di *Sprint* ini, dan kinerja sebelumnya dari *Development Team*. Jumlah pekerjaan yang dipilih dari *Product Backlog* untuk *Sprint* merupakan keputusan *Development Team* sepenuhnya. Hanya *Development Team* yang dapat menilai apa yang dapat mereka selesaikan di *Sprint*..[13]

4) *Sprint Goal*

Sprint Goal adalah sebuah obyektif untuk *Sprint* yang dapat dicapai lewat pengimplementasian *Product Backlog*. *Sprint Goal* merupakan panduan bagi *Development Team* untuk menjawab pertanyaan mengapa mereka mengembangkan Increment. *Sprint Goal* dibuat pada pertemuan *Sprint Planning*. *Sprint* memberikan ruang fleksibilitas mengenai fungsionalitas yang akan diimplementasikan di dalam *Sprint*. *Product Backlog* item terpilih merupakan satu fungsi yang selaras yang bisa jadi *Sprint Goal*. *Sprint Goal* bisa saja menghubungkan pekerjaan yang tidak memiliki keterkaitan agar *Development Team* tidak bekerja dari instruksi kerja yang berbeda-beda. *Development Team* selalu mengingat *Sprint Goal* yang telah ditetapkan pada saat mereka bekerja. *Development Team* mengimplementasikan fungsionalitas dan teknologi guna mencapai *Sprint Goal* tersebut. Apabila pekerjaannya ternyata berbeda dengan apa yang diharapkan oleh *Development Team* maka mereka akan berkolaborasi dengan *Product Owner* untuk menegosiasikan ruang lingkup *Sprint Backlog* di dalam *Sprint*. [14]

5) *Daily Scrum*

Kegiatan ini mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *Daily Scrum* sebelumnya dan melakukan prakiraan terhadap pekerjaan selanjutnya di dalam *Sprint*. *Daily Scrum* dilakukan di waktu dan tempat yang sama setiap harinya untuk mengurangi kompleksitas. *Development Team* menggunakan *Daily Scrum* untuk menginspeksi perkembangan

pekerjaan menuju Sprint Goal dan tren perkembangan penyelesaian pekerjaan di Sprint Backlog. Daily Scrum meningkatkan kemungkinan Development Team untuk mencapai Sprint Goal. Setiap hari, Development Team harus memahami bagaimana mereka bekerjasama sebagai tim swakelola untuk mencapai Sprint Goal dan membuat Increment yang diharapkan di akhir Sprint.[15]

Development Team atau beberapa anggota tim seringkali berkumpul setelah Daily Scrum untuk diskusi yang lebih mendalam atau melakukan adaptasi atau melakukan rencana ulang terhadap sisa pekerjaan dari Sprint. Daily Scrum adalah pertemuan internal untuk Development Team. Jika orang lain hadir, Scrum Master memastikan mereka tidak mengganggu jalannya pertemuan ini.[15]

6) *Sprint Review*

Sprint Review diselenggarakan di akhir Sprint untuk menginspeksi Increment dan mengadaptasi Product Backlog bila diperlukan. Pada saat Sprint Review, Scrum Team dan pemegang kepentingan berkolaborasi untuk meninjau apa yang sudah diselesaikan di Sprint. Berdasarkan hasil tinjauan tersebut dan perubahan terhadap Product Backlog di dalam Sprint, hadirin berkolaborasi untuk menentukan pekerjaan selanjutnya yang dapat dilakukan untuk mengoptimalkan nilai bisnis. Ini adalah pertemuan informal, bukan pertemuan laporan status, dan presentasi Increment dilakukan guna mendapatkan umpan balik dan mengembangkan kemampuan kolaborasi. [16]

H. *RFID*

RFID atau bisa disebut juga *Radio Frequency Identification* adalah sistem identifikasi berbasis wireless yang memungkinkan pengambilan data tanpa harus bersentuhan seperti *barcode* atau *magnetic card*.

RFID menggunakan sistem identifikasi dengan gelombang radio, karena itu minimal dibutuhkan dua buah perangkat agar alat ini dapat berfungsi, adapun perangkat yang dibutuhkan disebut *TAG* dan *READER*. [17]

II. METODE PENELITIAN

A. *Agile Development*

Metode yang digunakan dalam pengembangan sistem informasi *Monitoring Perkuliahan Umum Terintegrasi Universitas Sam Ratulangi Manado Daerah* adalah *Agile Development* dengan menggunakan yang merupakan *Framework Scrum*.

Dalam pengembangan sistem informasi monitoring perkuliahan terintegrasi Universitas Sam Ratulangi tahapan penerapan *Agile Development* dan menggunakan framework *Scrum* adalah sebagai berikut :

1). *User Stories*: Pada bagian ini Product Owner akan memberikan spesifikasi dan sesuai dengan kebutuhan oleh users. Dalam user stories berisi nama pengguna sistem, fitur-fitur yang menjadi kebutuhan sistem dan tujuan dari fitur yang direncanakan.

TABEL I.
PRODUCT BACKLOG

| No | Fitur | Estimasi (hari) | Status Prioritas |
|----|--|-----------------|------------------|
| 1. | Integrasi Template | 3 | Prioritas Utama |
| 2. | Integrasi <i>database user</i> untuk login menggunakan <i>username</i> dan <i>password</i> portal akademik | 3 | Prioritas Utama |
| 3. | Integrasi mesin <i>RFID</i> dengan aplikasi | 21 | Prioritas Utama |
| 4. | Integrasi dengan aplikasi RPS | 7 | Prioritas Sedang |
| 5. | Integrasi <i>database</i> kelas dengan aplikasi | 3 | Prioritas Utama |
| 6. | Pengelolaan data mata kuliah | 7 | Prioritas Utama |
| 7. | Pengelolaan data presensi mahasiswa | 7 | Prioritas Utama |
| 8. | Moitoring data presensi mahasiswa | 7 | Prioritas Utama |
| 9. | Pengelolaan data jurnal kelas | 7 | Prioritas Utama |

2). *Product Backlog*:

Pada tahap ini hasil dari user stories tersebut dibagi menjadi 3 bagian besar yaitu dosen , mahasiswa dan admin. Berdasarkan 3 bagian tersebut hanya 2 bagian besar fitur yaitu fitur dosen dan fitur operator atau admin dalam tabel ini akan di tunjukan mana yang menjadi prioritas fitur sesuai dengan kesepakatan antara Product Owner dan Development Team. Berikut tabel produk backlog

III. HASIL DAN PEMBAHASAN

A *Sprint 1*

Sprint akan dibagi menjadi 2 sprint berdasarkan product Backlog pada tabel 1 yaitu sprint 1 akan fokus dalam proses integrasi antar database unsrat ke database aplikasi , serta melakukan integrasi template tampilan agar sesuai dengan standart pengembangan aplikasi di Pusat Teknologi Informasi Universitas Sam Ratulangi Manado dan juga melakukan pengerjaan fitur absensi menggunakan RFID. Sprint 2 akan berfokus pada pengelolaan data aplikasi dan pengerjaan fungsi monitoring absensi, jurnal perkuliahan, daftar kelas dari dosen dan integrasi aplikasi RPS dengan aplikasi penulis.

1) *Sprint Planning*

Pada tahap sprint planning ini akan dipecah menjadi 3 tugas atau fungsi berdasarkan dari hasil product Backlog yaitu integrasi template tampilan, integrasi database dosen ,serta penggunaan absen menggunakan *RFID*.

TABEL II.

FITUR *BACKLOG* PADA SPRINT 1

| No | Fitur Backlog | Tugas | Estimasi | Durasi | Status |
|----|-----------------------------|--|----------|--------|---------|
| 1. | Integrasi Template tampilan | - Install composer php | 1 | 1 | Selesai |
| | | - Unduh template tikdev gitlab | 2 | 3 | Selesai |
| | | -Konfigurasi template menggunakan HMVC | 3 | 3 | Selesai |
| 2. | Integrasi database | - Unduh <i>sqllyog</i> | 1 | 1 | Selesai |
| | | - Memahami database di sitdev | 3 | 3 | Selesai |
| | | -Login sebagai dosen | 3 | 3 | Selesai |
| 3. | Absensi <i>RFID</i> | - Konfigurasi <i>control panel</i> alat <i>webpass</i> | 7 | 7 | Selesai |
| | | - <i>Cron Jobs</i> 1 dari alat <i>scanner</i> ke database <i>unsrat_monkul</i> iah <i>data_mesin</i> | 7 | 7 | Selesai |
| | | - <i>Cron Jobs</i> 2 Integrasi database <i>akademika_sia</i> <i>s_kelas</i> dengan <i>s_presensi_kel</i> <i>as_detil</i> | 7 | 7 | Selesai |

2) *Sprint Backlog*

Pada tahap sprint backlog, penulis akan merancang fitur serta flow aplikasi dalam bentuk flowchart atau Activity diagram yang menggambarkan kebutuhan sistem. Perancangan database juga dilakukan dalam sprint backlog yang bertujuan memudahkan para developer untuk mengikuti hasil dari perancangan yang diskusikan pada awal pengembangan aplikasi sehingga tidak membingungkan pada saat pengembangan berlangsung

3) *Sprint Execution dan Daily Scrum*

Pada tahap sprint execution penulis membuat tabel 1. untuk melihat progress pembuat aplikasi ini yang dijabarkan sesuai dengan product backlog pada sprint 1.

Pada tahap ini juga dilakukan testing atau pengujian dengan berbagai scenario terkait fitur fitur yang di kembangkan.

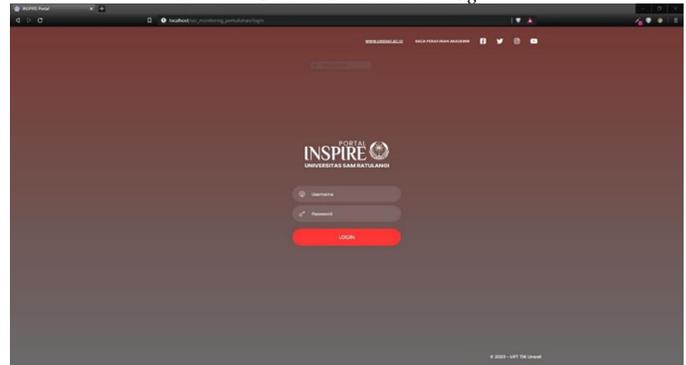
4) *Sprint review*

Ketika suatu fitur sudah selesai maka akan dilakukan demonstrasi fitur pada product owner untuk menanggapi fitur tersebut jika kekurangan atau penambahan fitur akan di diskusikan dan akan dimasukkan di product backlog pada sprint selanjutnya.

TABEL III.
TABEL III.

SKENARIO *TESTING* PADA SPRINT 1.

| No | Skenario Pengujian | Output | Validasi |
|----|--|--|----------|
| 1. | Dosen memasukan <i>username</i> dan <i>password</i> portal akademik | Masuk ke dalam halaman <i>dashboard</i> aplikasi | Sukses |
| 2. | Dosen salah memasukan <i>username</i> dan <i>password</i> | Validasi <i>error password</i> atau <i>usemame</i> salah | Sukses |
| 3. | Scan kartu <i>RFID</i> (cron 1) | NIM tersimpan di dalam storage di database | Sukses |
| 4. | Absensi <i>RFID</i> dosen membuka dan menutup absen sesuai dengan pertemuan (cron 2) | Hasil absensi tersimpan di database <i>akademika_sia</i> sesuai dengan pertemuan absensi | Sukses |
| 5. | Mahasiswa melakukan absensi tanpa dosen membuka absensi atau menutup (cron 2) | Hasil absensi tidak tersimpan di database <i>akademika_sia</i> | Sukses |
| 6. | Dosen tidak menutup absen (cron 2) | Hasil dari absen tersebut tersimpan ke dalam <i>akademika_sia</i> | Gagal |
| 7. | Mahasiswa melakukan absensi 2 x pernama (cron 2) | Mahasiswa tersebut absensinya hanya dihitung sebagai 1 x kehadiran | Sukses |

Gambar 5. Antarmuka *login*.5) *Sprint Retrospective*

Pada tahap ini akan dilakukan evaluasi selama proses sprint 1 berjalan, jika ada kekurangan dapat ditingkat pada sprint yang akan datang. Hasil dari tahap ini adalah terdapat banyak hambatan dimana salah satunya tidak tersediaya API yang menghubungkan alat scanner rfid dengan aplikasi ini maka penulis membuat script untuk mengambil data yang tersimpan dalam storage mesin ini dengan menggunakan teknik scraping dengan menggunakan bahasa pemograman PHP.

TABEL IV.
SKENARIO TESTING APLIKASI PADA SPRINT 2

| No | Skenario Pengujian | Output | Validasi |
|----|--|---|----------|
| 1. | Masuk menggunakan salah satu akun dosen | Kelas kelas yang diampuh oleh dosen tersebut akan muncul di halaman dashboard aplikasi | Sukses |
| 2. | Dosen mengakses fitur absensi | Jadwal pertemuan dan jumlah pertemuan serta mahasiswa yang mengkontak mata kuliah muncul | Sukses |
| 3. | Dosen melakukan absensi pada salah satu jadwal pertemuan | Mahasiswa yang mengkontrak mata kuliah tersebut namanya akan masuk dalam tabel kehadiran dan hasil absensi akan di masukan kedalam grafik per pertemuan | Sukses |
| 4. | Dosen melakukan absensi tidak pada jadwal pertemuan | Hasil absensi tersebut tidak akan tersimpan kedalam tabel kehadiran atau terhitung tidak valid | Sukses |
| 5. | Dosen melakukan pengisian rps pada aplikasi rps | Masuk pada jurnal perkuliahan sesuai dengan mata kuliah terkait | Sukses |
| 6. | Dosen menambahkan materi pada pertemuan | Materi akan tersimpan pada pertemuan terkait | Sukses |

B Sprint 2

1) Sprint Planning:

Pada tahap sprint planning pada sprint 2 ini akan berfokus pada integrasi aplikasi rps unsrat, monitoring perkuliahan , jurnal kelas , absensi mahasiswa.

2) Sprint Backlog

Pada tahap sprint backlog , penulis akan merancang fitur serta flow aplikasi dalam bentuk flowchart atau Activity diagram yang menggambarkan kebutuhan sistem serta melakukan perancangan database hingga perancangan interface.

3) Sprint Execution dan Daily Scrum

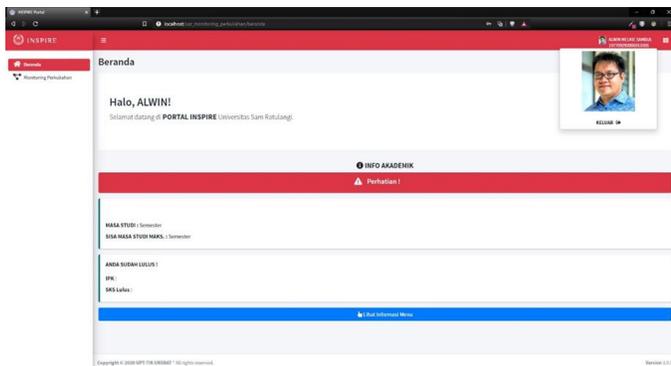
Pada tahap sprint execution penulis membuat tabel 4.3 untuk melihat progress pembuatan aplikasi ini yang dijabarkan sesuai dengan product backlog pada sprint 2

4) Sprint review

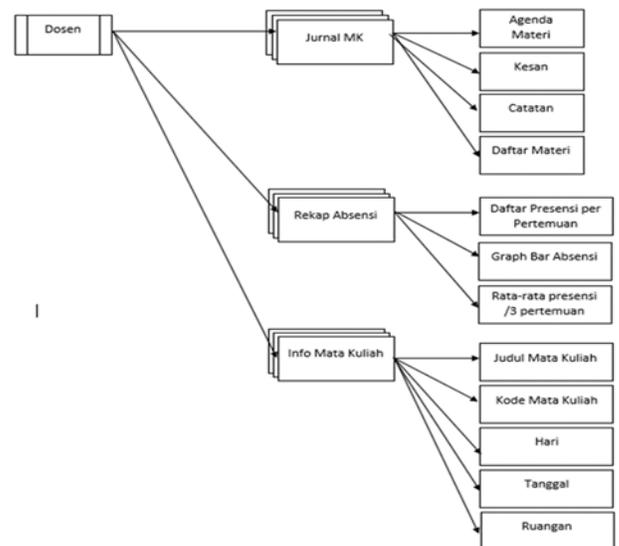
Ketika suatu fitur sudah selesai maka akan dilakukan demonstrasi fitur pada product owner untuk menanggapi fitur tersebut jika kekurangan atau penambahan fitur akan di diskusikan dan akan dimasukan di product backlog.

5) Sprint retrospective:

Pada tahap ini akan dilakukan evaluasi selama proses sprint 2 berjalan, jika ada kekurangan dapat ditingkat pada spirnt yang akan datang. Hasil dari tahap ini adalah fitur absensi bisa ditampilkan dalam bentuk interface aplikasi sehingga memudahkan dosen untuk melakukan monitoring absensi mahasiswa serta yang disajikan melalui tampilan grafik pada fitur absensi dosen dengan dapat melihat tingkat kehadiran mahasiswa berdasarkan pertemuan. Fitur jurnal perkuliahan dimaksud agar dosen dapat melakukan perencanaan yang lebih baik sehingga semester berlangsung bisa langsung melihat berdasarkan RPS yang terintegrasi dengan aplikasi penulis. Tantangan pada sprint 2 adalah kompleksitas struktur database dari Universitas Sam Ratulangi hingga menyulitkan penulis dalam mengembangkan aplikasi ini karena untuk mendapatkan suatu data yang dibutuhkan memerlukan teknik join pada database yang cukup banyak dan harus mengikuti standart yang diterapkan.



Gambar 6. Antarmuka *dashboard*.



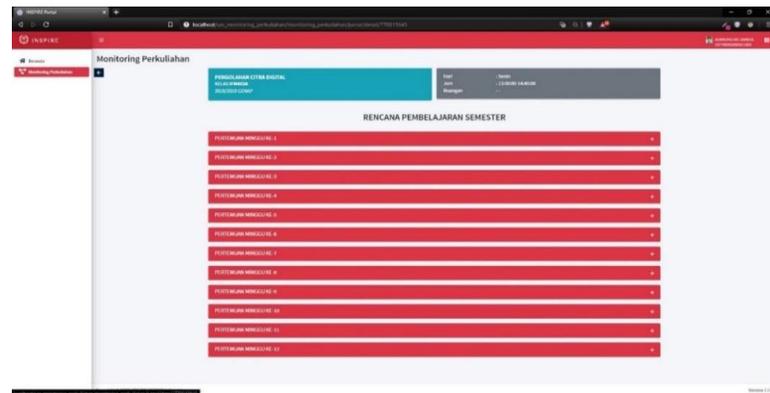
Gambar 7. Activity Diagram sprint 2.

TABEL V.
FITUR BACKLOG PADA SPRINT 2

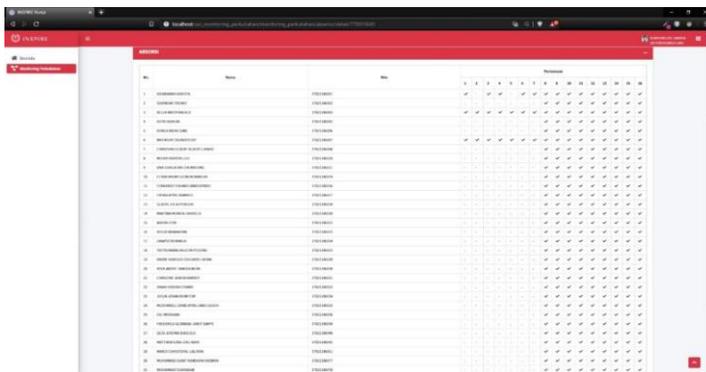
| No | Fitur Backlog | Tugas | Estimasi | Durasi | Status |
|----|--|--|----------|--------|---------|
| 1. | Integrasi database kelas dengan aplikasi | - Fungsi mendapatkan kelas , semester , kode kelas , jadwal kelas oleh dosen terkait | 5 | 7 | Selesai |
| | | - Desain antarmuka dashboard kelas – kelas yang di ampu | 2 | 7 | Selesai |
| 2. | Pengelolaan data presensi mahasiswa | - Fungsi untuk mendapatkan mahasiswa yang mengkontrak mata kuliah tersebut | 3 | 7 | Selesai |
| | | - Fungsi jadwal pertemuan per mata kuliah yang terkait | 3 | 7 | Selesai |
| | | - Desain tampilan absensi mahasiswa | 2 | 7 | Selesai |
| | | | | | |
| 3. | Monitoring data presensi kehadiran mahasiswa | - Grafik jumlah absensi jumlah mahasiswa yang hadir. | 5 | 7 | Selesai |
| | | - Fungsi 3 pertemuan terakhir | 2 | 7 | Selesai |
| 4. | Pengelolaan data jurnal kelas | - Integrasi dengan aplikasi rps | 3 | 7 | Selesai |
| | | - Input kesan mata kuliah | 2 | 7 | Selesai |
| | | - Input catatan per pertemuan | 2 | 7 | Selesai |
| | | - Input tugas | 1 | 7 | Selesai |



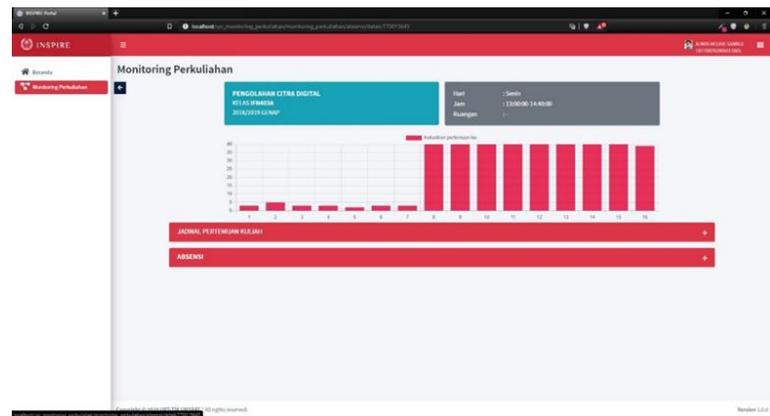
Gambar 9. Antarmuka kelas.



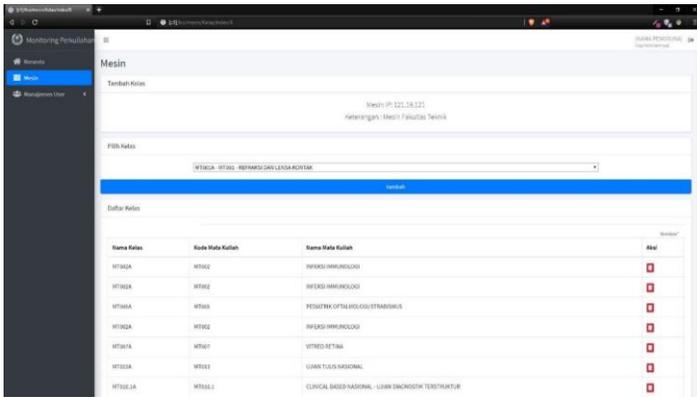
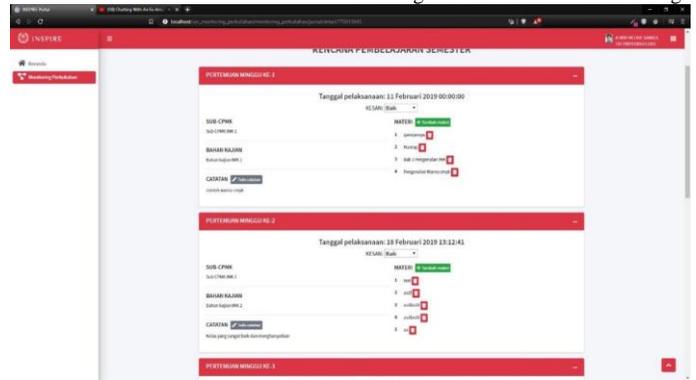
Gambar 10. Antarmuka monitoring perkuliahan



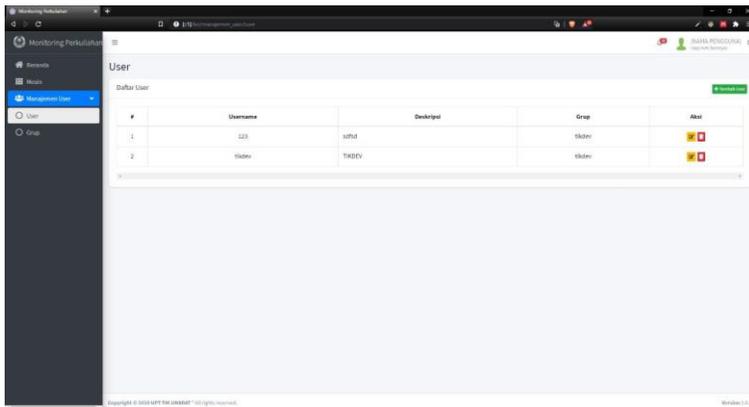
Gambar 8.. Antarmuka jadwal pertemuan kelas.



Gambar 11.. Antarmuka presensi mahasiswa

Gambar 12. Antarmuka *backoffice* penambahan kelas

Gambar 13. Antarmuka pertemuan perminggu dari rps

Gambar 14. Antarmuka manajemen *user*.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Telah dikembangkan Sistem informasi Monitoring perkuliahan terintegrasi universitas Sam Ratulangi Manado berbasis web yang dapat diakses secara online dan digunakan oleh dosen untuk melakukan monitoring perkuliahan dan Sistem Informasi ini telah terintegrasi dengan aplikasi RPS sehingga dosen dapat melakukan pengisian di aplikasi RPS dan menjadi jurnal perkuliahan di aplikasi ini. Aplikasi saat ini terhubung dengan aplikasi krs dan manajemen perkuliahan yang berada di universitas Sam Ratulangi.

B. Saran

Berdasarkan kesimpulan diatas, maka disarankan untuk melakukan pelatihan bagi dosen untuk menggunakan aplikasi ini sehingga absensi dalam berjalan dengan optimal. Dan juga melakukan integrasi dengan aplikasi lain sehingga fitur dari rps dan absensi bisa berjalan secara maksimal serta diharapkan pengembangan memperhatikan teknis penempatan mesin absensi yang rencanya akan di letakan di masing masing kelas sehingga alat tersebut dapat digunakan dengan baik baik serta terhindar dari kerusakan teknis.

V. KUTIPAN

- [1] Ependi, U. Implementasi Model Scrum pada Sistem Informasi Seleksi Masuk Mahasiswa Politeknik Pariwisata Palembang, 2018.
- [2] Andry, J. F. Sistem Informasi Monitoring Proyek Furniture di PT. XYZ. Jurnal Sistem Informasi, 2016.
- [3] Ibarani, et.al. Metode Scrum. Imperial Journal of Interdisciplinary Research (IJIR), 2017.
- [4] Fernando, D., Anharudin, A., & Fadli, F. Rancang Bangun Aplikasi E-Portofolio Hasil Karya Mahasiswa Unsera Menggunakan Metode Scrum, 2018.
- [5] Herliana A., & Rasyid, P. M. Sistem Informasi Monitoring Software Pada Tahap, 2018.
- [6] Hananto, V. R. RANCANG BANGUN SISTEM INFORMASI MONITORING PERKULIAHAN KELAS BERBASIS WEB (STUDI KASUS STIKOM SURABAYA) (Design Of Web-based Information System for Monitoring Class Lectures Case Study : STIKOM Surabaya , 2010.
- [7] Priyanto, W., Nugroho, D., & Widada, B. Sistem Informasi Monitoring Perkuliahan Berbasis Web di STMIK Sinar Nusantara Surakarta, 2015.
- [8] Ruseno, N. Implementasi Scrum pada Pengembangan Aplikasi Sistem Reservasi Online Menggunakan PHP, 2019
- [9] MUDJAHIDIN, M., & DITA PAHANG PUTRA, N. Rancang Bangun Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web, 2012.
- [10] Rezania Agramanisti. Implementasi Scrum Pada Pengembangan Software Distribusi, 2013.



Penulis bernama lengkap Geraldo Franciscus Thomas Jose Lantang anak ke pertama dari dua bersaudara, lahir di Manado pada tanggal 20 Juli 1998. Penulis menempuh pendidikan pertama di TK Santa Theresia Malalayang Manado 2002-2003, kemudian melanjutkan ke SD Santa Theresia Malalayang Manado pada tahun 2003-2009, setelah itu melanjutkan sekolah di SMP Katolik Pax Christi Manado pada tahun 2009-2012, kemudian melanjutkan pendidikan di SMA Katolik Rex Mundi Manado pada tahun 2012-2015. Tahun 2015, penulis melanjutkan studi di Fakultas Teknik, Jurusan Teknik Elektro Program Studi Teknik Informatika, Universitas Sam Ratulangi Manado. Penulis melaksanakan kerja praktek di Codepolitan Bandung Jawa Barat selama 3 bulan yaitu Juli – September 2018.

