

**PENGEMBANGAN SISTEM MONITORING MICROSOFT SQL  
SERVER (STUDI KASUS : IPB)**

**SHALHAN RADIFAN PILLI**



**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2017**



**PERNYATAAN MENGENAI SKRIPSI DAN  
SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA**

Dengan ini saya menyatakan bahwa skripsi berjudul Pengembangan Sistem Monitoring Microsoft SQL Server (Studi Kasus : IPB) adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, Desember 2017

*Shalhan Radifan Pilli*  
NIM G64130087

## ABSTRAK

SHALHAN RADIFAN PILLI. Pengembangan Sistem *Monitoring* Microsoft SQL Server (Studi Kasus : IPB). Dibimbing oleh IRMAN HERMADI

Institut Pertanian Bogor (IPB) sedang mengembangkan sistem informasi sebagai pendukung kegiatan perkuliahan. Seluruh pengadaan sistem informasi di IPB berada di bawah pengawasan Direktorat Integrasi Data dan Sistem Informasi (DIDSI) IPB. DIDSI bertanggung jawab terhadap komponen-komponen dari sistem informasi di IPB seperti server dan basisdata. Kestabilan komponen tersebut perlu terus dijaga agar sistem tetap berjalan dengan normal. Walaupun fitur *monitoring* basisdata telah ada pada *Database Management System* (DBMS) yang digunakan oleh DIDSI yaitu Microsoft SQL Server (MSSQL), namun hal tersebut tidak bisa diimplementasikan karena pengaksesan terhadap DBMS akan melanggar aturan yang ada. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem *monitoring* MSSQL. Sistem tersebut dapat mengawasi kondisi server basisdata dan juga aktivitas yang terjadi pada basisdata. Pengembangan sistem menggunakan bahasa pemrograman Java, Android SDK, dan metode *Prototyping*. Tahapan yang dilakukan dalam metode *Prototyping* adalah komunikasi, perencanaan cepat, perancangan cepat, pembuatan prototipe, dan evaluasi prototipe. Tahapan diulang dari awal jika setelah evaluasi masih perlu perbaikan ataupun pengembangan pada prototipe. .

Kata kunci: sistem monitoring, DBMS, *Prototyping*

## ABSTRACT

SHALHAN RADIFAN PILLI. Development Of Microsoft SQL Server Monitoring System (Case Study : IPB). Supervised by IRMAN HERMADI.

*Bogor Agricultural University is developing an information system to support learning activities. All the procurement of information system in Bogor Agricultural University is under supervision Directorates Data Integration and Information System. They are responsible for all of the information system component such as server and database. The stability of those components should be maintained in order to keep system running normaly. Although the native auditing tools is already exist in the Database Management System (DBMS) that used by Bogor Agricultural University which is Microsoft SQL Server (MSSQL), however using a native auditing tools is impossible because it will break the existing rules. Therefore, this study aim to develope a MSSQL monitoring system. The system can monitor server condition and database activites. The system is developed by using Java and Android SDK as the programming language, and Prototyping as the system development method. Prototyping method have five stages which are communication, quick plan, quick design, construction of prototype, and evaluation. The stages are repeated from beginning if improvement and development is still needed after doing an evaluation of the prototype.*

*Keywords: monitoring system, DBMS, Prototyping.*

**PENGEMBANGAN SISTEM MONITORING MICROSOFT  
SQL SERVER (STUDI KASUS : IPB)**

**SHALHAN RADIFAN PILLI**

Skripsi  
sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Komputer  
pada  
Departemen Ilmu Komputer

**DEPARTEMEN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
INSTITUT PERTANIAN BOGOR  
BOGOR  
2017**

|

Dikomentari [sr1]: Belum diubah

Penguji:  
1 Dean Apriana Ramadhan, SKomp MKom  
2 Firman Ardiansyah, SKom MSi

Judul Skripsi : Pengembangan Sistem *Monitoring* Microsoft SQL Server (Studi Kasus : IPB)  
Nama : Shalhan Radifan Pilli  
NIM : G64130087

Disetujui oleh

Irman Hermadi, SKom MS PhD  
Pembimbing

Diketahui oleh

Prof Dr Ir Agus Buono, MSi MKom  
Ketua Departemen

Tanggal Lulus :

## PRAKATA

Puji dan syukur kepada Allah *subhanahu wa ta'ala* atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Desember 2016 ini adalah sistem informasi, dengan judul Pengembangan Sistem *Monitoring* Microsoft SQL Server (Studi Kasus : IPB).

Banyak kendala dan tantangan dari awal penelitian dilaksanakan. Berkat doa, dukungan, dan kasih sayang dari Ibu Ratna Rosita, Ayah Januar, Kakak Nadia Amalia Pilli, dan Adik Nabil Biopari Pilli menjadi kekuatan tersendiri untuk menyelesaikan penelitian ini. Terima kasih juga penulis sampaikan kepada Bapak Irman Hermadi, SKom MS PhD selaku dosen pembimbing yang senantiasa membantu, mengarahkan, dan memberi solusi ketika terjadi kesulitan dan ikhlas memberikan waktunya untuk membimbing penulis selama penelitian. Penulis juga menyampaikan terima kasih kepada:

1. Bapak Dean Apriana Ramadhan, SKomp MKom dan Bapak Ahmad Ridha, SKom MS selaku penguji.
2. Seluruh dosen, staf tata usaha, dan staf pegawai Departemen Ilmu Komputer IPB.
3. Seluruh teman sebangkisan yang membantu hingga skripsi ini terselesaikan yaitu Alif, Dikna, Rizqi dan Hafiz
4. Teman-teman kontrakan dan teman-teman seperjuangan di Departemen Ilmu Komputer IPB yaitu Rafif, Aga, Alif, Arya, Wildan, Dipa, Bayulisar, dan Denny yang selalu memberikan semangat serta dukungannya.

Semoga segala bantuan, bimbingan, motivasi dan dukungan yang telah diberikan kepada penulis senantiasa dibalas oleh Allah *subhanahu wa ta'ala*. Semoga karya ilmiah ini memberikan dan wawasan kepada pembaca.

Bogor, Desember 2017

*Shalhan Radifan Pilli*



## DAFTAR ISI

ABSTRAK	iv
DAFTAR ISI	ix
DAFTAR TABEL	x
DAFTAR GAMBAR	x
DAFTAR LAMPIRAN	x
PENDAHULUAN	1
Latar Belakang	1
Perumusan Masalah	2
Tujuan Penelitian	2
Manfaat Penelitian	2
Ruang Lingkup Penelitian	2
METODE	2
Data Penelitian	2
Tahapan Penelitian	2
Lingkungan Pengembangan	4
HASIL DAN PEMBAHASAN	4
Iterasi Pertama	4
Iterasi Kedua	9
SIMPULAN DAN SARAN	14
Simpulan	14
Saran	15
DAFTAR PUSTAKA	15
LAMPIRAN	16
RIWAYAT HIDUP	29

## DAFTAR TABEL

1. Hasil analisis kebutuhan pengguna sistem monitoring MSSQL	5
2. Hasil analisis kebutuhan pengguna sistem monitoring MSSQL iterasi 2	10
3. Use case description all job info	16
4. Use case description monitoring hardware usage	17
5. Use case description all job info	18
6. Use case description add admin	19
7. Use case description save admin log	20

## DAFTAR GAMBAR

1. Metode Prototyping	3
2. Use case diagram sistem monitoring MSSQL	5
3. Flowchart sistem MSSQL	6
4. Activity diagram fungsi monitoring job	7
5. Class diagram sistem monitoring MSSQL	7
6. Sequence diagram monitoring job sistem monitoring MSSQL	8
7. Hasil prototipe sistem monitoring MSSQL iterasi 1	9
8. Use case diagram sistem monitoring MSSQL iterasi 2	11
9. Flowchart sistem monitoring MSSQL iterasi 2	11
10. Activity diagram fungsi add IP address sistem monitoring MSSQL	12
11. Sequence diagram use case add server sistem monitoring MSSQL	13
12. Hasil perancangan prototipe sistem monitoring MSSQL iterasi kedua	14

## DAFTAR LAMPIRAN

1. Use case description sistem monitoring MSSQL iterasi 1	16
2. Activity diagram sistem monitoring MSSQL Iterasi 1	22
3. Sequence diagram sistem monitoring MSSQL iterasi 1	24

## PENDAHULUAN

### Latar Belakang

Institut Pertanian Bogor (IPB) merupakan salah satu instansi yang menggunakan sistem informasi untuk menunjang kegiatan akademik institut yang dikelola oleh Direktorat Integrasi Data dan Sistem Informasi (DIDSI). Sebagai penanggung jawab sistem informasi di IPB, DIDSI berperan penting dalam mengawasi basisdata IPB. Basisdata berguna sebagai pengorganisasian data sehingga memudahkan pengguna dalam memperoleh informasi (Kadir 2014). Karena itu, basisdata merupakan komponen yang penting dalam penyelenggaraan sebuah sistem informasi.

Pengembangan basisdata pada dasarnya memerlukan *tools* yang sering disebut dengan *Database Management System* (DBMS). IPB menggunakan Microsoft SQL Server (MSSQL) sebagai DBMS institut. Berdasarkan data dari DB-Engine (2016), MSSQL berada di peringkat ketiga sebagai DBMS terpopuler di dunia dengan skor 1226. Peringkat tersebut berada di atas PostgreSQL dan MongoDB yang menempati peringkat empat dan lima. Selain IPB, MSSQL juga digunakan oleh perusahaan besar seperti Microsoft Corporation, Eagle Creek Software Service, dan Kiewit Corporation (Idatalabs 2016).

Sebagai sebuah DBMS yang digunakan untuk mengontrol laju informasi yang besar, MSSQL memiliki sebuah fitur *built-in* yang bernama Performance Monitoring and Tuning Tools yang berguna untuk mengawasi aktivitas basisdata. Fitur tersebut dapat memberikan informasi transaksi sql yang dilakukan oleh sistem dan juga dapat memberikan informasi pada level *hardware* seperti *cpu*, memori, dan *hardisk* (Microsoft 2008). Namun karena sifatnya yang *built-in*, maka fitur tersebut tidak cocok digunakan oleh perusahaan ataupun lembaga besar karena pada umumnya pengaksesan terhadap DBMS melanggar aturan dan data menjadi rawan untuk dimanipulasi (McAfee 2012). Karena itu dibutuhkan sebuah pendekatan lain untuk melakukan *monitoring* kerja basisdata.

Terdapat tiga pendekatan untuk *monitoring* aktivitas basisdata yaitu *native auditing*, *network-based monitoring*, dan *host-based agent* (Howart 2012). *Native auditing* adalah metode *monitoring* yang menggunakan fitur bawaan dari DBMS yang digunakan. *Network-based* merupakan metode *monitoring* dengan menggunakan alat (*hardware*) yang ditanamkan pada *port server* sehingga dapat menangkap informasi seperti *traffic* dan perubahan konfigurasi network yang melalui jaringan tersebut. *Host-based agent* adalah metode *monitoring* dengan menggunakan sebuah program yang telah terinstalasi di dalam server. Program tersebut yang bertugas memberikan ulasan informasi terkait aktivitas sistem secara keseluruhan mulai dari kondisi server hingga aktivitas basisdata. Namun pada umumnya program *monitoring host-based agent* memanfaatkan *native auditing* dari DBMS yang digunakan.

Lumbanraja pada tahun 2015 mengembangkan sistem *monitoring* server IPB dengan metode *Prototyping*. Sistem akan memberikan informasi *hardware* yang bekerja pada server seperti memori dan *hardisk* dan sistem dapat memberikan informasi *service* yang sedang aktif pada server, seperti HTTP, HTTPS, MySQL, dan SMTP. Jika terdeteksi adanya gangguan pada server maka sistem akan otomatis

mengirimkan notifikasi kerusakan melalui *email*. Namun, sistem tersebut masih belum mencakup *monitoring* aktivitas basisdata.

*Monitoring* terhadap basisdata adalah hal yang krusial agar aktivitas tidak normal pada data dapat ditangani dengan cepat (Dharani dan Sangeetha 2013). Karena itu, pengembangan sistem *monitoring* selanjutnya akan mencakup *monitoring* aktivitas basisdata dan menggunakan pendekatan *host-based agent* agar pengawasan terhadap perangkat keras pada server tetap dapat dilakukan. Sistem *monitoring* harus bisa memberikan informasi mengenai kondisi server basisdata secara *real-time* dan berkala.

### **Perumusan Masalah**

Berdasarkan latar belakang diatas, masalah yang diangkat dalam penelitian ini adalah bagaimana mengembangkan sebuah aplikasi Android untuk *monitoring* MSSQL yang tidak memerlukan akses langsung ke dalam MSSQL.

### **Tujuan Penelitian**

Tujuan penelitian ini adalah mengembangkan aplikasi Android yang dapat melakukan *monitoring* MSSQL berikut mesin servernya.

### **Manfaat Penelitian**

Manfaat dari penelitian ini adalah memberikan kemudahan dalam melakukan *monitoring* aktivitas server basisdata dan basisdata di IPB.

### **Ruang Lingkup Penelitian**

Ruang lingkup penelitian ini adalah indikator *monitoring* ditentukan berdasarkan kebutuhan dari Direktorat Integrasi Data dan Sistem Informasi (DIDSI) IPB.

## **METODE**

### **Data Penelitian**

Data yang digunakan dalam penelitian ini adalah parameter yang terdapat pada MSSQL seperti nama basisdata, nama *instance*, IP *address*, *username*, dan *password* basisdata. Data yang digunakan bersifat lokal karena pengembangan sistem ini masih dilakukan pada localhost

### **Tahapan Penelitian**

Metode pengembangan yang dilakukan adalah metode *Prototyping*. *Prototyping* adalah metode pengembangan perangkat lunak dimana *developer* (pengembang) menyelesaikan *prototype* (prototipe) secara cepat terlebih dahulu lalu melakukan pengujian secara berulang-ulang hingga mendapatkan perangkat

lunak yang sesuai dengan kebutuhan pengguna. Komunikasi langsung kepada *stakeholder* secara berkala merupakan kunci untuk menghasilkan perangkat lunak yang sesuai dengan kebutuhan. Metode ini cocok diterapkan ketika *stakeholder* hanya memahami tujuan umum dari aplikasi yang akan dikembangkan (Pressman 2010).

Pengembangan perangkat lunak dengan menggunakan metode *Prototyping* terdiri dari beberapa tahapan yaitu *communication*, *quick plan*, *modeling quick design*, *construction of prototype*, dan *deployment delivery & feedback*. Tahapan pada metode *Prototyping* dilakukan secara berurutan dan dilakukan pengulangan berdasarkan pada kebutuhan dan kepuasan pihak *stakeholder* terhadap perangkat lunak yang dikembangkan seperti yang ditunjukkan pada gambar 2 berikut.



Gambar 1 Metode *Prototyping*

#### **Communication (komunikasi)**

Metode Prototyping diawali dengan komunikasi langsung dengan DIDS sebagai *stakeholder*. Selama proses komunikasi, pengembang menganalisis kebutuhan perangkat lunak berdasarkan tujuan yang ingin dicapai oleh DIDS. Tak hanya itu, pengembang juga menganalisis proses bisnis dari perangkat lunak yang akan dikembangkan.

#### **Quick Plan (perencanaan cepat)**

Perencanaan dilakukan secara cepat berdasarkan kebutuhan dan keinginan DIDS. Pada tahap perencanaan cepat, fungsi yang dibutuhkan oleh sistem dari hasil komunikasi kepada DIDS dianalisis lebih lanjut untuk menggambarkan proses bisnisnya. Proses bisnis tersebut direpresentasikan dalam bentuk *flowchart* dan *activity diagram*.

#### **Modeling Quick Design (pemodelan rancangan cepat)**

Pemodelan dan rancangan cepat berfokus pada aspek dari sistem yang terlihat oleh user seperti masukan dan keluaran. Pada tahapan ini, dilakukan pemodelan perilaku sistem, struktur sistem dan perancangan antarmuka sistem. Pemodelan perilaku sistem dan struktur sistem direpresentasikan kedalam *sequence diagram* dan *class diagram*.

#### **Construction of Prototyping (pembuatan prototipe)**

Pembuatan prototipe dilakukan berdasarkan hasil dari perencanaan cepat

dan pemodelan rancangan cepat. Pembuatan prototipe bertujuan agar *stakeholder* dapat merasakan sistem yang diinginkan terlebih dahulu sebelum mempublikasikannya. Pembuatan prototipe juga digunakan sebagai mekanisme untuk mengidentifikasi kebutuhan sistem.

#### **Deployment Delivery & Feedback (penyebaran, pengiriman, dan umpan balik)**

Pada tahapan ini prototipe yang telah dikembangkan sebelumnya dievaluasi oleh DIDSi untuk mencari kelemahan maupun kekurangannya. Pengevaluasian dilakukan secara keseluruhan, mulai dari kelayakan tiap fungsi pada prototipe maupun *interface* prototipe. Setelah dilakukan evaluasi, DIDSi akan memberikan umpan balik ke pengembang. Jika diperlukan perbaikan pada sistem maka proses pengembangan kembali dari awal.

#### **Lingkungan Pengembangan**

Penelitian mengenai sistem *monitoring* MSSQL dilakukan dengan menggunakan perangkat keras dan perangkat lunak sebagai berikut.

- 1 Operating System: Windows 10 64 bit
- 2 Laptop dengan spesifikasi RAM 4096 MB dan Intel® Core™ i5-3317U CPU @ 1.70GHz× 4
- 3 Android Studio 2.2.3
- 4 Java 9
- 5 Android SDK 25.2.5
- 6 SQL Server 2016
- 7 Smartphone dengan OS Android Lollipop 5.0.2

### **HASIL DAN PEMBAHASAN**

#### **Iterasi Pertama**

##### **Komunikasi**

Pada iterasi pertama dilakukan komunikasi langsung kepada *stakeholder* yang dalam kasus ini adalah administrator basisdata di DIDSi IPB. Komunikasi dilakukan untuk mendapatkan permasalahan yang sedang dihadapi oleh administrator dalam melakukan *monitoring* basisdata. Kekurangan dan kelebihan dari penggunaan sistem yang sudah ada pun dapat diketahui. Ketiga faktor tersebut digunakan sebagai landasan untuk kebutuhan pengguna.

Hasil analisis kebutuhan pengguna untuk sistem *monitoring* MSSQL adalah sistem memiliki dua jenis pengguna, yaitu super admin dan admin. Admin dan super admin dapat menggunakan semua fungsi *monitoring* yang ada pada sistem. Namun super admin memiliki beberapa tambahan tugas yang tidak bisa dilakukan oleh admin biasa yaitu super admin dapat menentukan siapa saja yang bisa menjadi admin dan super admin juga bisa mengawasi aktivitas admin pada sistem *monitoring* MSSQL.

Fungsi *monitoring* yang dibutuhkan pengguna antara lain adalah *monitoring cpu usage*, *monitoring memory usage* dan *monitoring job* MSSQL. Job merupakan salah satu fitur yang ada di MSSQL yang berguna untuk melakukan *scheduling*

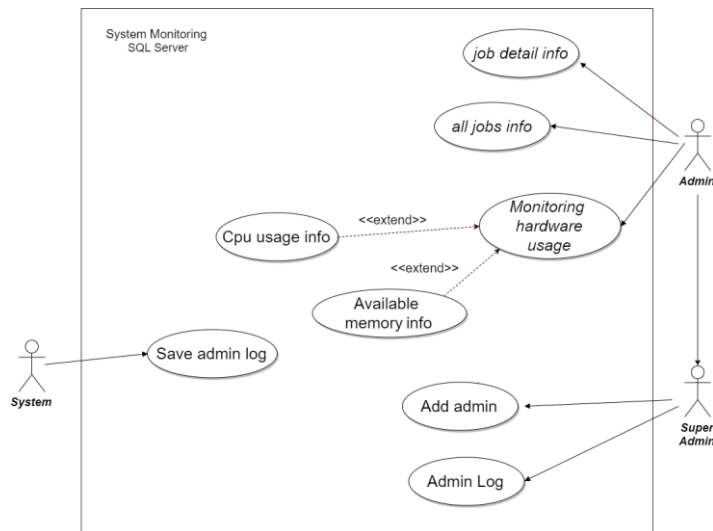
query. Parameter yang diambil antara lain adalah nama *job*, id, *last run time*, *last run date*, dan status.

Tabel 1 Hasil analisis kebutuhan pengguna sistem *monitoring* MSSQL

Pengguna	Kebutuhan pengguna
Super Admin	<i>Monitoring job</i> <i>Monitoring cpu usage</i> <i>Monitoring memory usage</i> Menambahkan admin <i>Monitoring aktivitas admin</i>
Admin	<i>Monitoring job</i> <i>Monitoring cpu usage</i> <i>Monitoring memory usage</i>

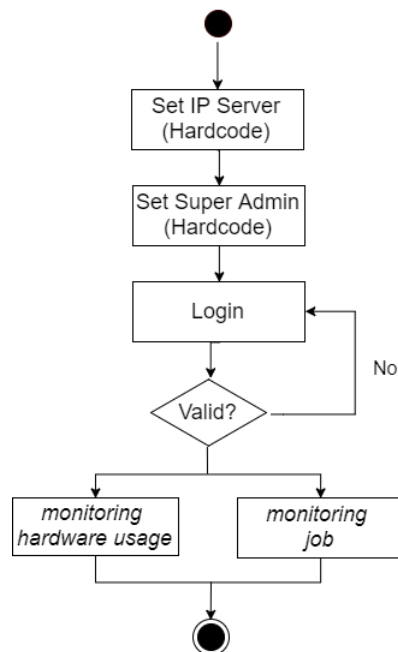
### Perencanaan Cepat

Hasil yang didapat dari tahapan komunikasi direpresentasikan kedalam bentuk *use case diagram* untuk menggambarkan kebutuhan sistem. Terdapat tiga aktor dan enam aktivitas utama pada *use case* yang dihasilkan. Aktor-aktor yang terlibat antara lain adalah admin, super admin, dan sistem. Sedangkan aktivitas yang ada antara lain adalah menampilkan *job info*, *job detail*, *cpu usage*, *available memory*, menambahkan admin, melihat *log* admin, dan menyimpan *log* admin. Seperti yang telah tertera pada gambar 2 dibawah ini, super admin memiliki akses keseluruhan fungsi yang ada pada sistem MSSQL. Deskripsi dari masing-masing *use case* direpresentasikan dalam *use case description* yang dapat dilihat pada Lampiran 1.



Gambar 2 Use case diagram sistem *monitoring* MSSQL

Kebutuhan sistem tersebut kemudian dianalisis lebih lanjut untuk mendapatkan alur kerja secara umum maupun alur kerja tiap fungsi yang dimiliki sistem. Alur kerja secara umum direpresentasikan ke dalam sebuah *flowchart*, sedangkan alur kerja tiap fungsi sistem direpresentasikan ke dalam sebuah *activity diagram*.

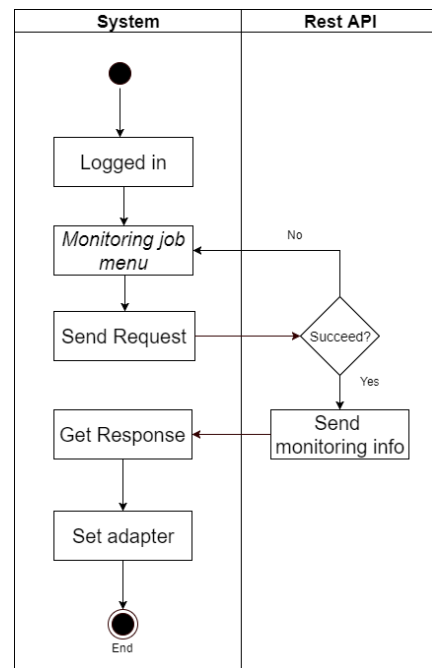


Gambar 3 *Flowchart* sistem MSSQL

Gambar 3 merupakan alur kerja dari sistem *monitoring* MSSQL. Untuk mendapatkan hasil secepatnya, pada iterasi satu ini pemasukan data super admin dan IP Address dari server yang ingin diawasi masih dilakukan secara *hardcode*. Jadi data super admin dan server yang ingin diawasi sudah ditentukan terlebih dahulu pada *script* sebelum sistem di *deploy*. Setelah sistem di *deploy* super admin dapat melakukan *login* pada sistem. Jika *login* berhasil maka super admin dapat melakukan *monitoring* server dengan fungsi-fungsi yang telah ditentukan pada tahapan sebelumnya.

Berdasarkan hasil dari tahapan komunikasi, proses *monitoring* sistem MSSQL terbagi dua yaitu *monitoring hardware usage* dan *monitoring job*. *Monitoring* bisa dilakukan bila admin telah melakukan *login*. Setelah melakukan *login*, admin diberikan pilihan menu *monitoring hardware usage* dan *monitoring job* yang bisa dipilih sendiri oleh admin. Kedua menu tersebut akan mengambil data dari MSSQL API. Detail dari salah satu proses *monitoring* direpresentasikan ke dalam bentuk *activity diagram* yang dapat dilihat pada gambar 4. Keseluruhan *activity diagram* sistem *monitoring* MSSQL dapat dilihat pada Lampiran 2.





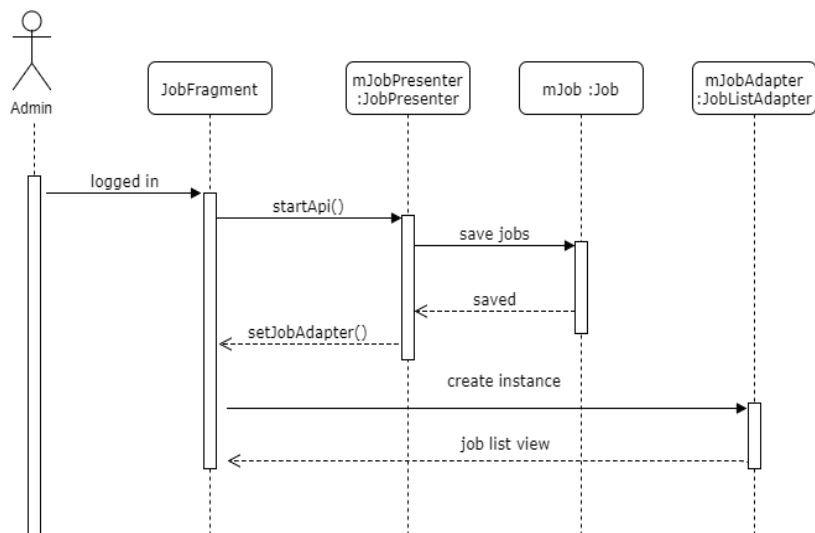
Gambar 4 Activity diagram fungsi monitoring job

### Perancangan Cepat

Tahap perancangan cepat dilakukan untuk mendeskripsikan kebutuhan yang diperlukan oleh sistem berdasarkan hasil analisis dari perencanaan cepat. Tahapan ini diawali dengan perancangan *database* (basisdata) sistem. Rancangan basisdata dalam pengembangan aplikasi berbasis *mobile* dibuat kedalam bentuk *class diagram*. Sistem *monitoring* MSSQL memiliki lima *class* yaitu *Job*, *JobDetail*, *Usage*, *User*, dan *UserLog*. Kelima *class* tersebut disesuaikan dari data yang diambil pada penelitian dari Dikna Romtika Yahdi yang berjudul Pengembangan *Representational State Transfer Application Program Interface* (REST API) Sistem *Monitoring* Microsoft SQL Server (Studi Kasus : IPB). *Class diagram* tersebut dapat dilihat pada gambar 5 dibawah ini.

Gambar 5 Class diagram sistem monitoring MSSQL

Rancangan yang dibutuhkan selanjutnya adalah *Sequence diagram*. Diagram ini berguna untuk menggambarkan interaksi antar objek pada sistem yang terbentuk dari kelas-kelas pada *class diagram*. Setiap *sequence* memiliki tiga kelas yaitu, *model*, *view*, dan *presenter* karena sistem menggunakan *architectural pattern* MVP. Dalam pengembangan sistem berbasis *android*, pada umumnya *view* dapat direpresentasikan dengan *fragment/activity*.

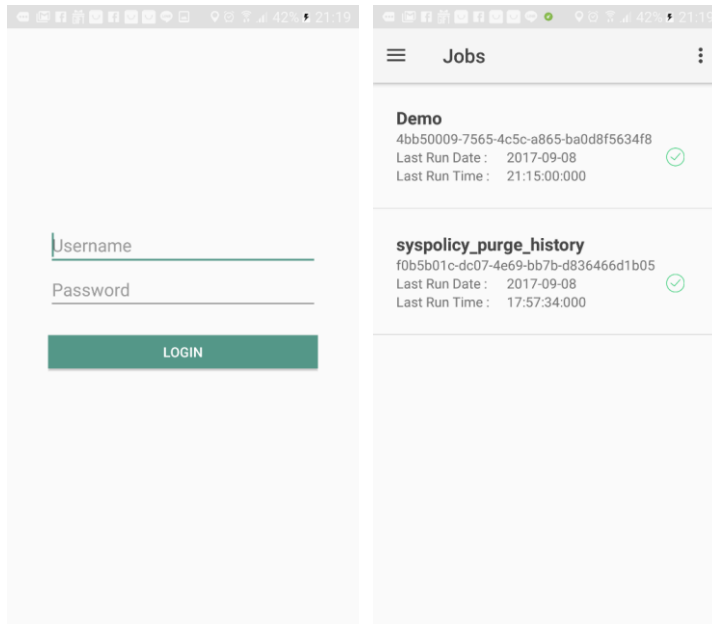


Gambar 6 Sequence diagram monitoring job sistem monitoring MSSQL

Gambar 6 merupakan salah satu *sequence diagram* untuk fungsi *monitoring job*. Aktor yang berperan dalam *monitoring job* adalah admin atau super admin. Aktor hanya dapat melihat informasi *job*. Untuk melihat informasi *job*, aktor diharuskan *login* terlebih dahulu lalu *JobActivity* mengirimkan perintah ke *JobPresenter* untuk menjalankan API. Jika *request* ke API berhasil, *JobPresenter* akan mendapatkan nilai dari informasi *job* yang mana akan disimpan ke dalam model *Job*. Setelah itu *JobPresenter* menjalankan fungsi *setJobAdapter* yang mana fungsi tersebut akan membuat *instance* dari *JobListAdapter* sehingga sistem dapat menampilkan informasi *job* kedalam bentuk *list view*. Keseluruhan dari *sequence diagram* sistem monitoring MSSQL dapat dilihat pada Lampiran 3.

### Perancangan Prototipe

Hasil dari perancangan yang telah dilakukan pada tahapan sebelumnya diimplementasikan kedalam *script* sehingga menghasilkan prototipe sesuai dengan yang diinginkan. Prototipe merupakan sistem yang berbasis *mobile* dengan OS Android 6.0 dan dikembangkan menggunakan bahasa Java dan *library* Android SDK. Beberapa gambar dari hasil antar muka dari sistem *monitoring MSSQL* dapat dilihat pada gambar 7 di bawah ini.



Gambar 7 Hasil prototipe sistem *monitoring* MSSQL iterasi 1

### Penyebaran, Pengiriman, dan Umpan Balik

Pada tahapan ini prototipe diserahkan kepada *stakeholder* untuk diuji setiap fungsinya. Pengujian dilakukan dengan menggunakan metode *black-box*. Metode *black-box* adalah metode pengujian aplikasi perangkat lunak yang memperhatikan *input* dan *output* dari setiap fungsi yang dijalankan. Hasil pengujian prototipe pertama seluruh fungsi berjalan dengan baik.

Setelah pengujian selesai dilakukan, selanjutnya *stakeholder* memberikan hasil evaluasinya. Berdasarkan hasil dari uji fungsi sistem yang dilakukan, *stakeholder* memberikan tambahan kebutuhan sistem untuk dapat melihat informasi *harddisk*, *longest job execution*, admin yang sedang login, fungsi untuk memasukan IP server secara dinamis sehingga pengguna dapat menentukan server mana saja yang ingin diawasi serta notifikasi bila ada server yang mati. Selain itu, *stakeholder* juga menginginkan adanya penambahan logo IPB di menu *login* dan menyesuaikan warna aplikasi dengan warna sistem IPB pada umumnya. Oleh karena itu, penelitian dilanjutkan ke iterasi kedua untuk memenuhi *requirement* dari *stakeholder*. Tabel hasil pengujian sistem disajikan pada Lampiran 4.

### Iterasi Kedua

#### Komunikasi

Komunikasi dengan *stakeholder* dilakukan kembali untuk mendapatkan *requirement* tambahan yang diperlukan. Berdasarkan hasil komunikasi, dari segi fungsi *stakeholder* menginginkan agar IP *address* server dapat dimasukkan oleh super admin, adanya informasi mengenai *longest duration job*, informasi admin

yang sedang *online*, informasi mengenai sisa *harddisk* dan notifikasi bila ada server yang mati. Sedangkan dari segi *interface*, *stakeholder* menginginkan adanya tambahan logo di halaman login sistem dan merubah warna pada komponen sistem menjadi biru tua sesuai dengan warna logo IPB. Tabel 2 menunjukkan seluruh kebutuhan pengguna sistem *monitoring* MSSQL iterasi 2.

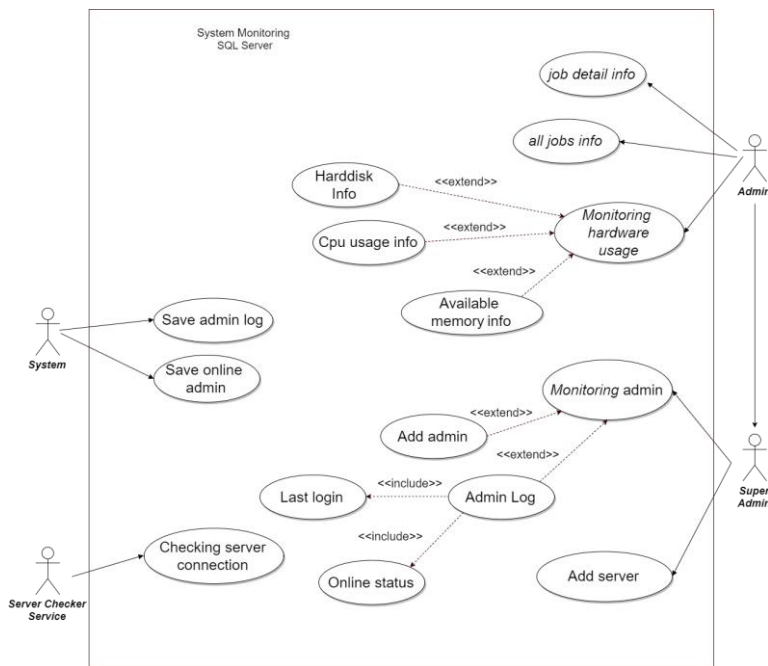
Tabel 2 Hasil analisis kebutuhan pengguna sistem *monitoring* MSSQL iterasi 2

Pengguna	Kebutuhan pengguna
Super Admin	<i>Monitoring job</i>
	<i>Monitoring cpu usage</i>
	<i>Monitoring memory usage</i>
	<i>Monitoring sisa harddisk</i>
	Informasi <i>longest job execution</i>
	Menambahkan admin
	Online admin
	<i>Monitoring aktivitas admin</i>
	Notifikasi server mati
	<i>Monitoring job</i>
Admin	<i>Monitoring cpu usage</i>
	<i>Monitoring memory usage</i>
	<i>Monitoring sisa harddisk</i>
	Informasi <i>longest job execution</i>
	Notifikasi server mati

### Perencanaan Cepat

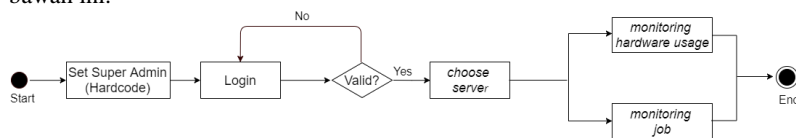
Kebutuhan pengguna yang telah didapatkan pada tahapan komunikasi dianalisis dan direpresentasikan kedalam bentuk *use case diagram* yang dapat dilihat pada gambar 8 di bawah ini. Pengembang memutuskan untuk tidak menambahkan *use case longest job execution* karena hasil dari *all job info* akan diurutkan berdasarkan durasi terbesar. Informasi status online *admin* juga akan digabungkan ke dalam admin log sehingga admin log nantinya akan menampilkan informasi *last login* serta status *online*.

Untuk mendapatkan status server yang aktif, pengembang memutuskan untuk membuat *service* baru yang berguna untuk mengecek semua status server dari *list* server pada sistem setiap menitnya. Jika ada server yang tidak aktif maka *service* tersebut akan mengirimkan notifikasi ke sistem. Pemisahan *service* tersebut dilakukan karena pengecekan server perlu dilakukan secara *real-time* sehingga menuntut sebuah sistem untuk bekerja secara terus menerus. Jika hal ini dilakukan oleh sistem utama (*mobile app*), maka akan membebani pengguna dari segi *resource*.



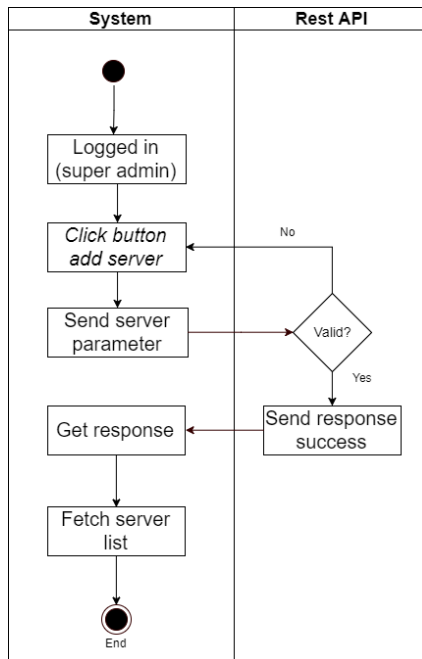
Gambar 8 Use case diagram sistem monitoring MSSQL iterasi 2

Karena adanya penambahan *use case*, maka alur proses sistem berubah. Admin harus memilih atau super admin harus menambahkan server yang ingin diawasi terlebih dahulu sebelum dapat melakukan *monitoring*. Alur proses sistem secara keseluruhan digambarkan ke dalam bentuk *flowchart* seperti gambar 9 di bawah ini.



Gambar 9 Flowchart sistem monitoring MSSQL iterasi 2

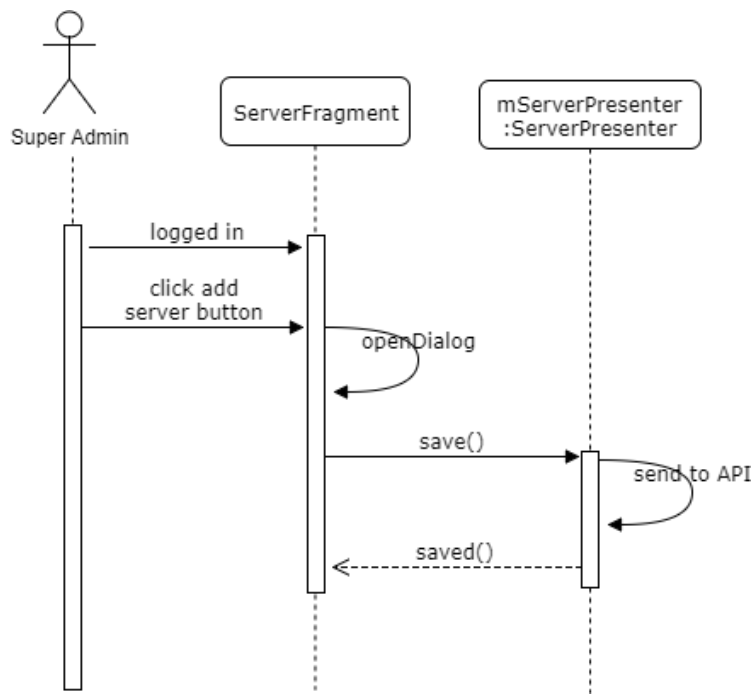
Penambahan server hanya dapat dilakukan oleh super admin dengan cara mengirimkan parameter yang dibutuhkan ke API sistem *monitoring* MSSQL yaitu IP address, username MSSQL dan password MSSQL. Jika parameter *valid* maka API akan mengirimkan *response success* dan admin maupun super admin dapat melakukan *monitoring* terhadap server tersebut. Semua tahapan tersebut digambarkan ke dalam bentuk *activity diagram* yang dapat dilihat pada gambar 10 di bawah ini. Keseluruhan *activity diagram* untuk setiap fungsi tambahan pada sistem *monitoring* MSSQL iterasi kedua dapat dilihat pada Lampiran 5.



Gambar 10 Activity diagram fungsi add IP address sistem monitoring MSSQL

### Perancangan Cepat

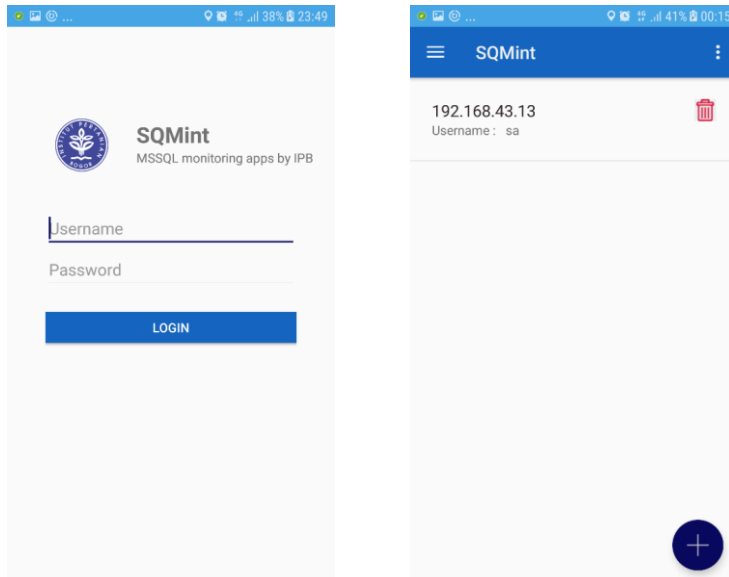
Pada tahapan ini, hasil dari perencanaan cepat dianalisis untuk memodelkan kebutuhan sistem yang direpresentasikan kedalam bentuk *sequence diagram*. Seperti yang terlihat pada gambar 11 dibawah ini, *class* yang diperlukan ada dua yaitu *serverfragment* dan *serverpresenter*. *ServerFragment* akan mengirimkan perintah *save server* ke *ServerPresenter* yang mana setelah itu *ServerPresenter* akan mengirimkan data inputan dari admin ke MSSQL API. Jika berhasil data akan tersimpan ke dalam MSSQL API. Keseluruhan *sequence diagram* untuk semua fungsi tambahan pada sistem *monitoring MSSQL* iterasi kedua dapat dilihat pada Lampiran 6.



Gambar 11 Sequence diagram use case add server sistem monitoring MSSQL

### Perancangan Prototipe

Prototipe yang telah dikerjakan di iterasi satu dikembangkan lagi sesuai dengan *requirement* terbaru dari *stakeholder* di iterasi kedua ini. Terdapat penambahan *requirement* dari segi fungsi dan segi *interface*. Dari segi fungsi, *stakeholder* menginginkan adanya informasi mengenai *available harddisk*, informasi *longest job execution*, informasi admin yang sedang *online*, dan penginputan *IP address* server secara dinamis. Sedangkan dari segi *interface* *stakeholder* menginginkan penambahan logo dihalaman *login* dan menyesuaikan warna komponen dari sistem dengan warna IPB pada umumnya yaitu biru tua. Beberapa hasil dari perancangan prototipe iterasi dua dapat dilihat pada gambar 12 di bawah ini. Keseluruhan gambar dari hasil *interface* sistem *monitoring* MSSQL dapat dilihat pada Lampiran 7.



Gambar 12 Hasil perancangan prototipe sistem *monitoring* MSSQL iterasi kedua

#### Penyebaran, Pengiriman, dan Umpan Balik

Pada tahapan ini prototipe diserahkan lagi kepada *stakeholder* untuk diuji setiap fungsinya. Pengujian tidak hanya dilakukan terhadap fungsi-fungsi yang baru saja dikembangkan di iterasi kedua, namun juga fungsi sistem secara keseluruhan. Pengujian masih menggunakan metode *black-box*. Hasil dari pengujian prototipe kedua ini antara lain adalah semua fungsi yang ada pada prototipe pertama dan semua fungsi tambahan pada iterasi kedua berjalan dengan baik. Hasil dari pengujian dapat dilihat pada Lampiran 8.

## SIMPULAN DAN SARAN

### Simpulan

Sistem *monitoring* MSSQL berbasis android berhasil dikembangkan dengan menggunakan metode *prototyping*. Sistem tersebut dapat melakukan *monitoring job* dan *monitoring hardware usage* sehingga pengawasan dapat dilakukan dimana saja. Admin juga tidak perlu mengakses langsung kedalam MSSQL Management (DBMS) sehingga keamanan data jadi lebih terjaga. Selain melakukan *monitoring* pada MSSQL, sistem juga dapat melakukan *monitoring* aktivitas admin.



### Saran

Sistem *monitoring* MSSQL masih perlu pengembangan lebih lanjut. Sistem hanya dapat digunakan pada sistem operasi Android yang mana dibutuhkan pengembangan agar sistem dapat digunakan di sistem operasi lain seperti IOS atau Windows. Keamanan pada sistem juga masih belum menjadi fokus utama pada penelitian ini sehingga kemungkinan masih adanya *bug* masih cukup besar. Selain itu, perlu adanya penelitian lebih lanjut yang berfokus pada *user experience* sehingga sistem jadi lebih mudah dimengerti oleh pengguna.

### DAFTAR PUSTAKA

- DB-Engines. 2016. DB-Ranking [Internet]. [diakses 2016 Desember 3]. Tersedia pada: <http://db-engines.com/en/ranking>.
- Dharani M, Sangeetha T. 2013. *Improving Policy Based Intrusion Response Component of a Relational Database by Evaluation of User*. *IJCTT - International Journal of Computer Trends and Technology*. 4(5), 1115-1117.
- Horwarth, J. 2012. *Setting Up a Database Security Logging and Monitoring Program* [Internet]. *SANS Institute: InfoSec Reading Room*. Tersedia pada: <https://www.sans.org/reading-room/whitepapers/application/setting-database-security-logging-monitoring-program-34222>
- Idatalabs. 2016. *Companies Using Microsoft SQL Server* [Internet]. [diakses 2016 Desember 3]. Tersedia pada: <https://idatalabs.com/tech/products/microsoft-sql-server>.
- Kadir A. 2014. *Pengenalan Sistem Informasi edisi Revisi*. Yogyakarta (ID) : ANDI.
- Lumbanraja R. 2015. *Pengembangan Sistem Monitoring Server Studi Kasus Institut Pertanian Bogor* [skripsi]. Bogor (ID): Institut Pertanian Bogor.
- McAfee. 2012. *McAfee Real-Time Database Monitoring, Auditing, and Intrusion Prevention* [Internet]. [diakses 2016 Desember 6]. Tersedia pada: <https://www.ascent.tech/wp-content/uploads/documents/mcafee/real-time-database-monitoring-auditing-intrusion-prevention.pdf>
- MSDN Microsoft. 2008. *Performance Monitoring and Tuning Tools* [Internet]. [diakses 2016 Desember 4]. Tersedia pada: [https://msdn.microsoft.com/en-us/library/ee377023\(v=bts.10\).aspx](https://msdn.microsoft.com/en-us/library/ee377023(v=bts.10).aspx)
- Pressman RS, 2010. *Software Engineering Practioner's Apparoach*. 7th ed. New York (US): McGrawHill.

## LAMPIRAN

Lampiran 1 *Use case description* sistem monitoring MSSQL iterasi 1

Tabel 3 *Use case description all job info*

<i>Use case name :</i>	<i>All job info</i>
<i>Scenario :</i>	Menampilkan informasi semua <i>job</i> berupa : <i>last date execution, last time execution, last duration, dan last status.</i>
<i>Triggering event :</i>	Masuk ke halaman <i>monitoring job</i>
<i>Brief description :</i>	<i>Monitoring job</i> dilakukan oleh admin dan super admin setelah melakukan login.
<i>Actors :</i>	Admin dan super admin
<i>Related use case :</i>	
<i>Stakeholders :</i>	Admin dan super admin
<i>Preconditions :</i>	Informasi <i>job</i> belum ditampilkan
<i>Postconditions :</i>	Informasi <i>job</i> berhasil ditampilkan
<i>Flow of activities :</i>	Alur aktivitas dapat dilihat pada gambar 4
<i>Exception conditions :</i>	

## Lampiran 1 Lanjutan

Tabel 4 *Use case description monitoring hardware usage*

<i>Use case name :</i>	<i>Monitoring hardware usage</i>
<i>Scenario :</i>	Menampilkan informasi memory dan cpu ke dalam bentuk chart.
<i>Triggering event :</i>	Masuk ke menu <i>monitoring usage</i>
<i>Brief description :</i>	<i>Monitoring usage</i> dilakukan oleh admin dan super admin setelah melakukan login dan masuk ke menu <i>monitoring usage</i> .
<i>Actors :</i>	Admin dan super admin
<i>Related use case :</i>	Extend : cpu usage info Extend : available memory info
<i>Stakeholders :</i>	Admin dan super admin
<i>Preconditions :</i>	Informasi <i>hardware usage</i> belum ditampilkan
<i>Postconditions :</i>	Informasi <i>hardware usage</i> berhasil ditampilkan ke dalam bentuk <i>chart</i>
<i>Flow of activities :</i>	Alur aktivitas dapat dilihat pada Lampiran 2.C
<i>Exception conditions :</i>	

## Lampiran 1 Lanjutan

Tabel 5 *Use case description all job info*

<i>Use case name :</i>	<i>All job info</i>
<i>Scenario :</i>	Menampilkan informasi semua <i>job</i> berupa : <i>last date execution</i> , <i>last time execution</i> , <i>last duration</i> , dan <i>last status</i> .
<i>Triggering event :</i>	Masuk ke halaman <i>monitoring job</i>
<i>Brief description :</i>	<i>Monitoring job</i> dilakukan oleh admin dan super admin setelah melakukan login.
<i>Actors :</i>	Admin dan super admin
<i>Related use case :</i>	
<i>Stakeholders :</i>	Admin dan super admin
<i>Preconditions :</i>	Informasi <i>job</i> belum ditampilkan
<i>Postconditions :</i>	Informasi <i>job</i> berhasil ditampilkan
<i>Flow of activities :</i>	Alur aktivitas dapat dilihat pada gambar 4
<i>Exception conditions :</i>	

## Lampiran 1 Lanjutan

Tabel 6 *Use case description add admin*

<i>Use case name :</i>	<i>Add admin</i>
<i>Scenario :</i>	Menambahkan admin agar bisa melakukan <i>monitoring</i>
<i>Triggering event :</i>	Masuk ke halaman admin lalu klik tombol <i>add admin</i>
<i>Brief description :</i>	<i>Add admin</i> dilakukan oleh super admin untuk memberikan <i>privilege</i> kepada seseorang agar bisa melakukan <i>monitoring</i>
<i>Actors :</i>	Super admin
<i>Related use case :</i>	
<i>Stakeholders :</i>	Super admin
<i>Preconditions :</i>	Admin belum ditambahkan
<i>Postconditions :</i>	Admin berhasil ditambahkan
<i>Flow of activities :</i>	Alur aktivitas dapat dilihat pada Lampiran 2.A
<i>Exception conditions :</i>	

## Lampiran 1 Lanjutan

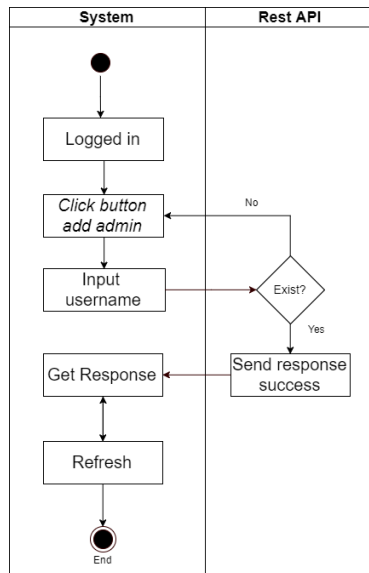
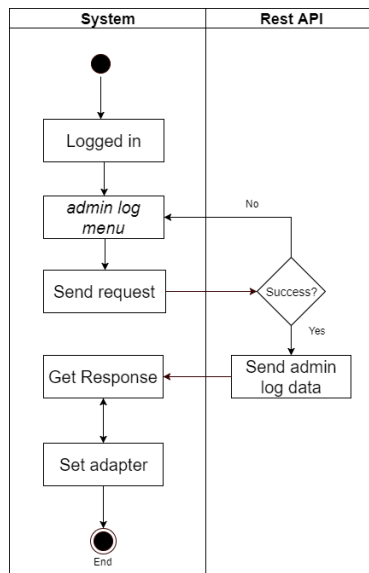
Tabel 7 *Use case description save admin log*

<i>Use case name :</i>	<i>Show admin log</i>
<i>Scenario :</i>	Super admin melakukan login lalu membuka halaman admin
<i>Triggering event :</i>	Saat super admin masuk ke halaman admin
<i>Brief description :</i>	<i>Show admin log</i> menampilkan semua list admin beserta tanggal dan waktu <i>last login</i> admin.
<i>Actors :</i>	Super admin
<i>Related use case :</i>	
<i>Stakeholders :</i>	Super admin
<i>Preconditions :</i>	Super admin belum masuk ke halaman admin
<i>Postconditions :</i>	Super admin berhasil melihat list admin beserta tanggal dan waktu <i>last login</i>
<i>Flow of activities :</i>	Alur aktivitas dapat dilihat pada Lampiran 2.B
<i>Exception conditions :</i>	

## Lampiran 1 Lanjutan

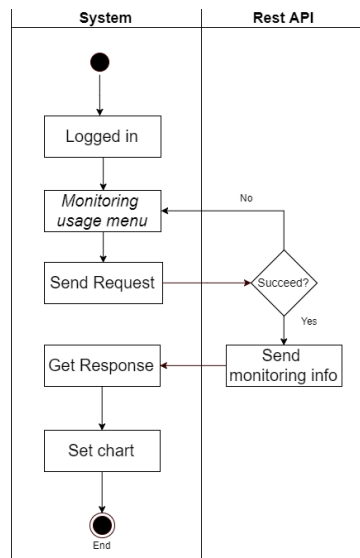
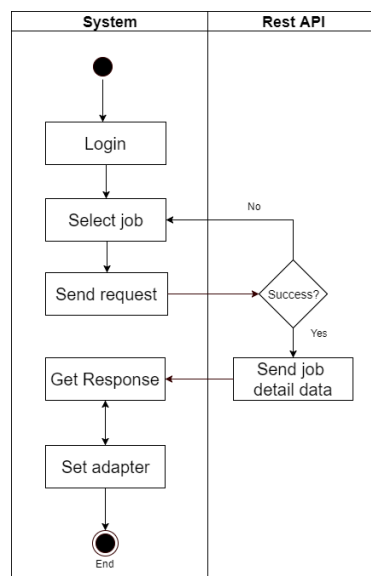
Tabel 8 *Use case description show admin log*

<i>Use case name :</i>	<i>Save admin log</i>
<i>Scenario :</i>	Saat super admin membuka halaman admin
<i>Triggering event :</i>	Masuk ke halaman login dan berhasil melakukan login.
<i>Brief description :</i>	<i>Save admin log</i> dilakukan oleh sistem setiap kali admin atau super admin melakukan <i>login</i> . Sistem akan menyimpan tanggal dan waktu saat admin melakukan login
<i>Actors :</i>	Sistem
<i>Related use case :</i>	
<i>Stakeholders :</i>	
<i>Preconditions :</i>	Admin atau super admin belum melakukan login
<i>Postconditions :</i>	Tanggal dan waktu admin melakukan login berhasil tersimpan
<i>Flow of activities :</i>	
<i>Exception conditions :</i>	

Lampiran 2 *Activity diagram sistem monitoring MSSQL Iterasi 1*A. *Add admin*B. *Admin log*

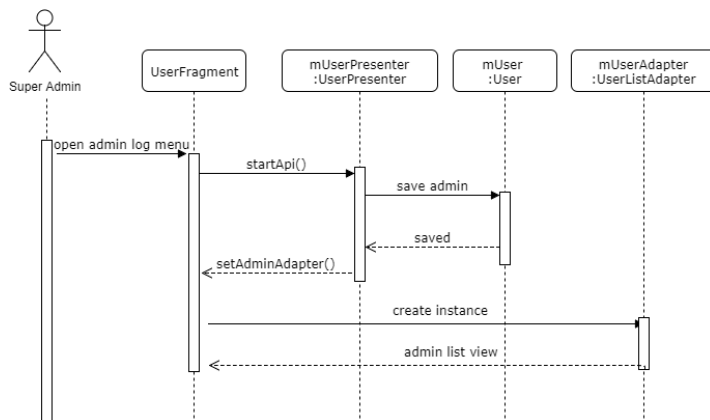


## Lampiran 2 Lanjutan

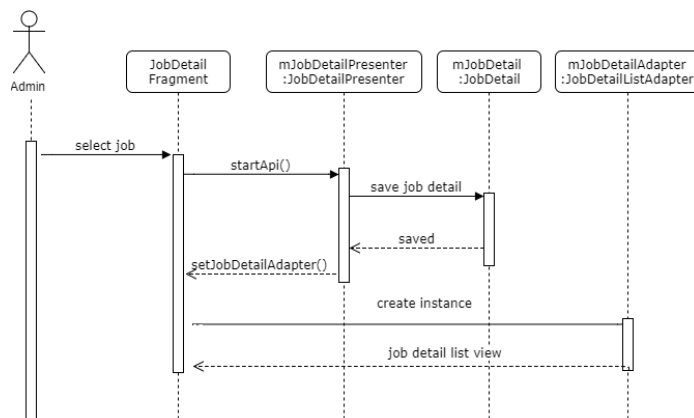
C. *Monitoring usage*D. *Job detail*

Lampiran 3 *Sequence diagram sistem monitoring MSSQL iterasi 1*

## A. Admin log

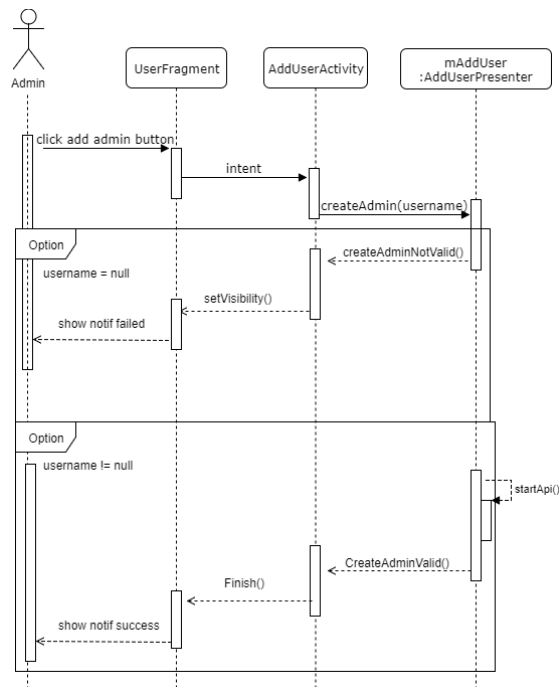


## B. Job detail

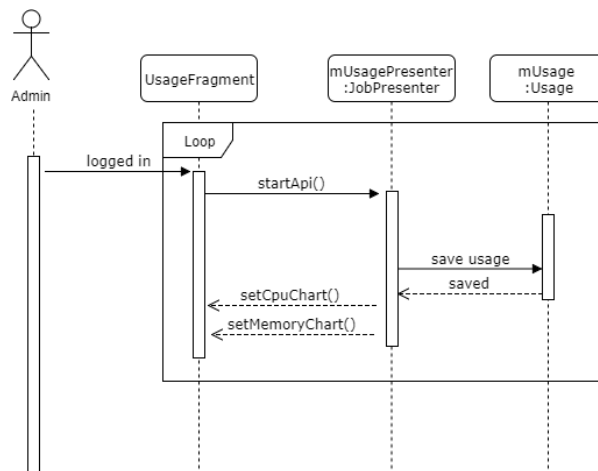


## Lampiran 3 Lanjutan

## C. Add admin



## D. Monitoring usage

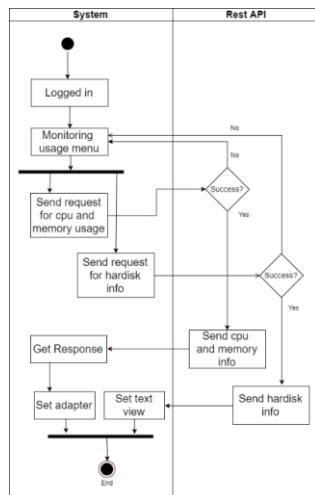


Lampiran 4 Hasil uji *black-box* iterasi 1Tabel 9 Hasil uji *black-box* sistem *monitoring* MSSQL iterasi 1

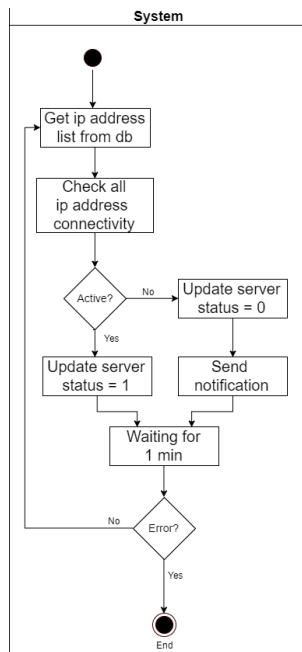
No	Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil uji
1	Melihat <i>job list</i>	Memilih menu <i>monitoring job</i>	Menampilkan <i>job list</i> beserta <i>last run date</i> , <i>last run time</i> , <i>last duration</i> , dan <i>last status</i> dari masing-masing <i>job</i>	Berhasil
2	Melihat <i>cpu usage</i> dan <i>memory usage</i>	Memilih menu <i>monitoring usage</i>	Menampilkan <i>cpu usage</i> dan <i>memory usage</i> kedalam bentuk <i>chart</i>	Berhasil
3	Melihat <i>job detail</i>	Memilih salah satu dari <i>job list</i>	Menampilkan <i>job detail</i> dari <i>job</i> yang dipilih dengan informasi berupa <i>run date</i> , <i>run time</i> , <i>duration</i> , dan <i>status</i>	Berhasil
4	Menambahk an admin	Super admin memasukkan username yang ingin dijadikan admin	Username berhasil ditambahkan kedalam daftar admin	Berhasil
5	Melihat admin <i>log</i>	Super admin memilih menu admin log	Menampilkan admin <i>list</i> beserta <i>last time login</i> dan <i>last date login</i> .	Berhasil

Lampiran 5 Activity diagram dari fungsi tambahan sistem *monitoring* MSSQL iterasi 2

#### A. Monitoring usage

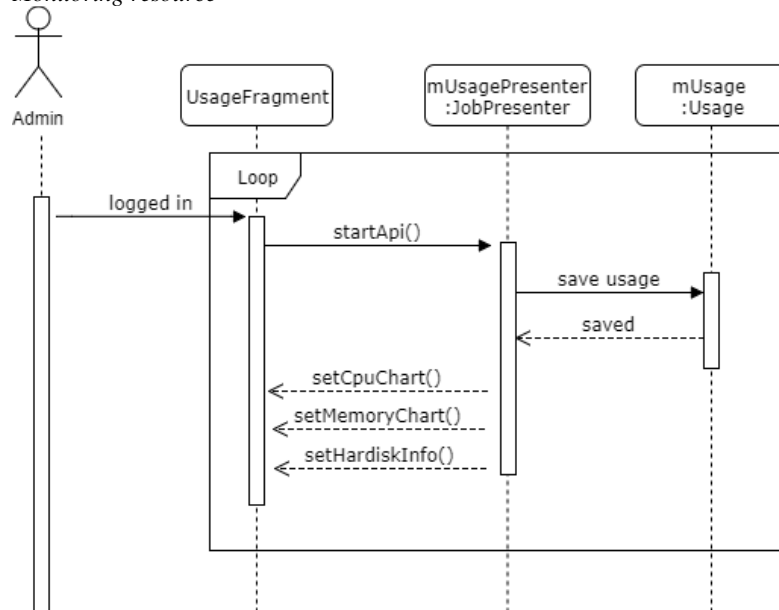


#### B. Server checker service



Lampiran 6 *Sequence diagram* fungsi tambahan sistem *monitoring* MSSQL iterasi 2

A. *Monitoring resource*



Lampiran 7 Hasil *prototipe* sistem *monitoring* MSSQL iterasi 2

Lampiran 8 Hasil uji *black-box* sistem *monitoring* MSSQL iterasi 2

Tabel 10 Hasil uji *black-box* sistem *monitoring* MSSQL iterasi 1

No	Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil uji
1	Melihat <i>job list</i>	Memilih server lalu memilih menu <i>monitoring job</i>	Menampilkan <i>job list</i> beserta <i>last run date</i> , <i>last run time</i> , <i>last duration</i> , dan <i>last status</i> dari masing-masing <i>job</i>	Berhasil
2	Melihat <i>cpu usage</i> dan <i>memory usage</i>	Memilih server lalu memilih menu <i>monitoring usage</i>	Menampilkan <i>cpu usage</i> dan <i>memory usage</i> kedalam bentuk <i>chart</i> dan	Berhasil

No	Pengujian	Skenario pengujian	Hasil yang diharapkan	Hasil uji
			menampilkan info <i>harddisk</i>	
3	Melihat <i>job detail</i>	Memilih salah satu dari <i>job list</i>	Menampilkan <i>job detail</i> dari <i>job</i> yang dipilih dengan informasi berupa <i>run date</i> , <i>run time</i> , <i>duration</i> , dan <i>status</i>	Berhasil
4	Menambahk an admin	Super admin <i>login</i> dan memasukkan username yang ingin dijadikan admin	Username berhasil ditambahkan kedalam daftar admin	Berhasil
5	Melihat admin <i>log</i>	Super admin <i>login</i> dan memilih menu admin log	Menampilkan admin <i>list</i> beserta <i>last time login</i> , <i>last date login</i> , dan <i>online status</i>	Berhasil
6	Menambahk an server	Super admin memasukkan parameter <i>ip address</i> , <i>username</i> , dan <i>password</i> MSSQL.	Menampilkan <i>alert success</i> dan server berhasil ditambahkan kedalam server <i>list</i>	Berhasil
7	Notifikasi server yang mati	Mematikan salah satu server dari server <i>list</i>	Menampilkan notifikasi	Berhasil

## RIWAYAT HIDUP

Penulis dilahirkan di Semarang pada tanggal 4 April 1995. Penulis adalah anak kedua dari ayah Ir Januar dan Ibu Ratna Rosita. Pada tahun 2013, penulis lulus dari SMA Negeri 1 Medan dan pada tahun yang sama, penulis lulus seleksi masuk Institut Pertanian Bogor (IPB) melalui jalur SBMPTN dan diterima di Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam IPB.

Selama kuliah, penulis aktif di organisasi Organisasi Mahasiswa Daerah (OMDA) Ikatan Mahasiswa Muslim Asal Medan (IMMAM) sebagai wakil kepala divisi SDM pada periode kepengurusan 2014/2015. Penulis juga pernah mengikuti

kepanitiaan “IT Today 2014” sebagai anggota divisi *Sponsorship*. Penulis melaksanakan Praktik Kerja Lapang (PKL) di departemen Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Institut Pertanian Bogor (IPB) pada tahun 2016 sebagai *web developer*.