

STUDY CASE CHAPTER 1
FUNDAMENTALS OF SOFTWARE ARCHITECT



Alifio Yudhistira Aji Salis

Cindi Dwi Pramudita

Soal no 2

Following are the 8 main functional requirements and 6 main non-functional requirements for the website <https://netmonk.id/>:

Use Case :

1. Adding a subscription (UC01)
2. View network status (UC02)
3. Manage network configuration (UC03)
4. Manage integration with other platforms (UC04)
5. Add user (UC05)
6. View analysis report (UC06)
7. View history of events (UC07)
8. View the list of subscriptions (UC08)

Use Case

ID	UC Name	UC Description
UC01	Adding Subscriptions	This use case describes how a network administrator adds a subscription or subscription service to gain access to features in Netmonk.
UC02	View Network Status	This use case describes how network administrators view network status on the Netmonk dashboard.
UC03	Manage Network Configuration	This use case describes how a network administrator manages network configuration using the features available on Netmonk.
UC04	Managing Integration with Other Platforms	This use case describes how network administrators manage integration with other platforms by using the features available on Netmonk.
UC05	Adding Users	This use case describes how a network administrator adds users to Netmonk.
UC06	View the Analytics Report	This use case describes how network administrators view the reports provided by Netmonk.
UC07	View Event History	This use case describes how network administrators view the event history and network usage provided by Netmonk.
UC08	View a list of subscriptions	This use case describes how a network administrator views the list of subscriptions that have been added

Kebutuhan Fungsional

ID	Statement	UC	Priority
F01	Memungkinkan Administrator Jaringan untuk menambahkan subscription, memvalidasi informasi subscription, dan menyimpan informasi subscription di database.	UC01	Must
F02	Memungkinkan Administrator Jaringan untuk melihat status jaringan secara real-time, termasuk ketersediaan perangkat jaringan dan kinerja jaringan dan menampilkan status jaringan dalam format yang mudah dibaca.	UC02	Must
F03	<ul style="list-style-type: none"> Memungkinkan Administrator Jaringan untuk mengelola konfigurasi jaringan seperti perangkat dan topologi jaringan. Memungkinkan Administrator Jaringan untuk menambahkan, mengubah, atau menghapus perangkat dan topologi jaringan dan menyimpan informasi konfigurasi jaringan di database. 	UC03	Must
F04	<ul style="list-style-type: none"> Memungkinkan Administrator Jaringan untuk mengintegrasikan Netmonk dengan platform lain seperti Slack, Microsoft Teams, atau layanan help desk lainnya. Memvalidasi informasi integrasi dengan platform lain dan menyimpan informasi integrasi dengan platform lain di database. 	UC04	Must
F05	Memungkinkan administrator jaringan untuk menambahkan pengguna baru, memvalidasi informasi pengguna dan menyimpan informasi pengguna di database.	UC05	Must
F06	<ul style="list-style-type: none"> Memungkinkan Administrator Jaringan untuk melihat laporan analisis jaringan, termasuk pemantauan kinerja dan keamanan jaringan Memungkinkan Administrator Jaringan untuk memilih rentang waktu laporan analisis dan menampilkan laporan analisis dalam format yang mudah dibaca. 	UC06	Must
F07	<ul style="list-style-type: none"> Memungkinkan Administrator Jaringan untuk melihat riwayat peristiwa jaringan, termasuk pemberitahuan masalah dan tindakan yang diambil Memungkinkan Administrator Jaringan untuk memilih rentang waktu riwayat peristiwa dan 	UC07	Must

	menampilkan riwayat peristiwa dalam format yang mudah dibaca.		
F08	Memungkinkan administrator untuk melihat daftar subscription yang telah ditambahkan sebelumnya.	UC08	Must

Kebutuhan Non Fungsional

ID	Pernyataan	Quality	UC	Priority
NF01	Netmonk harus memiliki sistem keamanan yang kuat dan dapat menjamin bahwa data yang dikumpulkan dan disimpan di dalam aplikasi Netmonk aman dan terjaga kerahasiaannya. Oleh karena itu, Netmonk harus memiliki fitur otentikasi dan otorisasi yang kuat dan juga enkripsi data yang disimpan di dalam aplikasi.	Data Security	UC01, UC03, UC05, UC06, UC07, UC08	Must
NF02	Aplikasi Netmonk harus tersedia sepanjang waktu untuk menghindari terjadinya downtime pada jaringan yang sedang dipantau. Ketersediaan ini harus dijamin oleh sistem yang handal dan juga redundansi yang baik.	Availability	UC02	Must
NF03	Netmonk harus dapat menghadapi banyak jumlah pengguna sekaligus dan dapat mengakomodasi permintaan yang tinggi dari pengguna dengan tetap menjaga performa yang stabil dan terukur.	Performance	UC02, UC06	Must
NF04	Netmonk harus dapat berkembang seiring dengan bertambahnya jumlah perangkat dan topologi jaringan yang terhubung ke dalamnya. Oleh karena itu, Netmonk harus mampu menangani kenaikan jumlah perangkat yang terhubung ke dalam aplikasi tanpa mengurangi kinerja dan fungsionalitasnya.	Scalability	UC02, UC03, UC06	Must
NF05	Netmonk harus mampu menjamin keamanan jaringan pengguna yang terhubung ke dalam aplikasi Netmonk. Oleh karena itu, Netmonk harus memiliki fitur keamanan yang kuat, seperti firewall dan enkripsi data yang dapat melindungi jaringan dari serangan dan ancaman yang dapat merusak keamanan jaringan.	Network Security	UC03, UC04, UC05, UC06, UC08	Must

NF06	Netmonk harus mematuhi standar keamanan dan regulasi yang berlaku seperti standar HIPAA, GDPR dan lain-lain.	Obedience	UC03, UC04, UC05, UC06	Must
------	--	-----------	---------------------------------	------

Soal no 3

1. User Story: Sebagai Administrator Jaringan, saya ingin menambahkan subscription baru dan memvalidasi informasi subscription untuk memonitor jaringan saya dengan lebih baik.

- Acceptance Criteria:

1. Sebagai Administrator Jaringan, saya harus dapat mengakses fitur untuk menambahkan subscription baru di dashboard Netmonk.
2. Saat menambahkan subscription, saya harus dapat memasukkan informasi yang dibutuhkan, seperti nama subscription, alamat IP jaringan yang akan dimonitor, dan informasi credential yang diperlukan.
3. Sistem harus dapat memvalidasi informasi yang dimasukkan oleh pengguna, seperti memastikan bahwa alamat IP valid dan credential benar.
4. Setelah informasi subscription diverifikasi, sistem harus dapat menyimpan informasi tersebut di database dengan aman.
5. Pengguna harus dapat melihat daftar subscription yang sudah ditambahkan dan menghapus subscription jika diperlukan.
6. Jika ada kesalahan atau masalah saat menambahkan subscription, sistem harus memberikan pesan error yang jelas untuk membantu pengguna memperbaiki masalah tersebut.

2. User Story: Sebagai Administrator Jaringan, saya ingin dapat melihat status jaringan secara real-time, sehingga saya dapat memantau ketersediaan perangkat jaringan dan kinerja jaringan dengan mudah.

- Acceptance Criteria:

1. Halaman status jaringan harus dapat menampilkan informasi tentang ketersediaan perangkat jaringan dan kinerja jaringan dalam waktu nyata.
2. Informasi yang ditampilkan harus mudah dipahami dan terstruktur dengan baik.
3. Pengguna harus dapat memperbarui status jaringan secara otomatis tanpa harus memuat ulang halaman.
4. Pengguna harus dapat memfilter informasi status jaringan berdasarkan kategori seperti perangkat jaringan, jenis perangkat, atau lokasi.
5. Pengguna harus dapat memperoleh notifikasi tentang masalah jaringan melalui fitur pemberitahuan atau notifikasi secara real-time.

3. User Story: Sebagai Administrator Jaringan, saya ingin dapat mengelola konfigurasi jaringan seperti perangkat dan topologi jaringan sehingga saya dapat menambahkan, mengubah, atau menghapus perangkat dan topologi jaringan dan menyimpan informasi konfigurasi jaringan di database.

- Acceptance Criteria:

1. Pengguna dapat menambahkan perangkat jaringan baru dan memasukkan informasi seperti nama, jenis, alamat IP, dan lokasi fisik.

2. Pengguna dapat mengubah informasi perangkat jaringan yang sudah ada seperti nama, jenis, alamat IP, dan lokasi fisik.
3. Pengguna dapat menghapus perangkat jaringan yang tidak diperlukan lagi dari database.
4. Pengguna dapat menambahkan topologi jaringan baru dan memasukkan informasi seperti nama, jenis, dan koneksi dengan perangkat jaringan lainnya.
5. Pengguna dapat mengubah informasi topologi jaringan yang sudah ada seperti nama, jenis, dan koneksi dengan perangkat jaringan lainnya.
6. Pengguna dapat menghapus topologi jaringan yang tidak diperlukan lagi dari database.
7. Setiap perubahan yang dilakukan harus disimpan di database dan dapat dilihat kembali pada saat yang dibutuhkan.

4. User Story: Sebagai Administrator Jaringan, saya ingin dapat mengintegrasikan Netmonk dengan platform lain seperti Slack, Microsoft Teams, atau layanan helpdesk lainnya, sehingga saya dapat menerima notifikasi dan informasi jaringan langsung melalui platform yang sudah saya gunakan sebelumnya.

- Acceptance Criteria:

1. Pengguna dapat memilih platform yang ingin diintegrasikan dengan Netmonk.
2. Pengguna harus memasukkan informasi yang valid untuk mengintegrasikan Netmonk dengan platform lain.
3. Sistem akan memvalidasi informasi integrasi dan memberikan notifikasi jika informasi yang dimasukkan tidak valid.
4. Setelah informasi integrasi berhasil divalidasi, sistem akan menyimpan informasi integrasi dengan platform lain di database.
5. Pengguna dapat mengelola dan mengubah informasi integrasi yang sudah disimpan di database.

5. User Story: Sebagai administrator jaringan, saya ingin dapat menambahkan pengguna baru ke Netmonk agar dapat memberikan akses kepada anggota tim saya untuk mengakses fitur monitoring jaringan.

- Acceptance Criteria:

1. Saya dapat menambahkan pengguna baru melalui form yang tersedia di Netmonk.
2. Setiap pengguna yang ditambahkan harus memiliki informasi akun yang valid, seperti nama pengguna, email, dan password yang kompleks.
3. Informasi pengguna baru harus disimpan dengan aman di dalam database.
4. Setelah pengguna berhasil ditambahkan, pengguna baru dapat masuk ke Netmonk dan mengakses fitur monitoring jaringan yang diperbolehkan oleh administrator.

6. User Story: Sebagai seorang Administrator Jaringan, saya ingin dapat melihat laporan analisis jaringan, termasuk pemantauan kinerja dan keamanan jaringan, sehingga saya dapat memantau kondisi jaringan secara keseluruhan dan membuat keputusan yang tepat.

- Acceptance Criteria:

1. Halaman laporan analisis jaringan dapat diakses oleh Administrator Jaringan setelah login ke Netmonk.
2. Administrator Jaringan dapat memilih rentang waktu laporan analisis yang ingin dilihat.
3. Laporan analisis jaringan termasuk informasi pemantauan kinerja dan keamanan jaringan.

4. Laporan analisis jaringan ditampilkan dalam format yang mudah dibaca dan dapat diunduh sebagai file.
5. Informasi laporan analisis jaringan bersifat terverifikasi dan akurat.

7. User Story: Sebagai Administrator Jaringan, saya ingin melihat riwayat peristiwa jaringan untuk memantau dan mengevaluasi performa jaringan, serta untuk mengetahui tindakan yang telah diambil untuk memperbaiki masalah.

- Acceptance Criteria:

1. Sistem menyediakan halaman riwayat peristiwa jaringan yang dapat diakses oleh Administrator Jaringan.
2. Administrator Jaringan dapat memilih rentang waktu riwayat peristiwa yang ingin ditampilkan.
3. Sistem menampilkan riwayat peristiwa dalam format yang mudah dibaca dan dapat dimengerti oleh Administrator Jaringan.
4. Informasi yang ditampilkan pada riwayat peristiwa mencakup tanggal dan waktu peristiwa terjadi, deskripsi singkat peristiwa, status peristiwa, tindakan yang diambil untuk memperbaiki masalah, dan pelapor peristiwa.
5. Riwayat peristiwa jaringan diambil dari data yang tersimpan di database dan diupdate secara real-time ketika terjadi peristiwa baru.

8. User Story: Sebagai Administrator Jaringan, saya ingin dapat melihat daftar subscription yang sudah ditambahkan dan detail informasi subscription yang sudah tersimpan, agar saya dapat memantau jaringan saya dengan lebih baik.

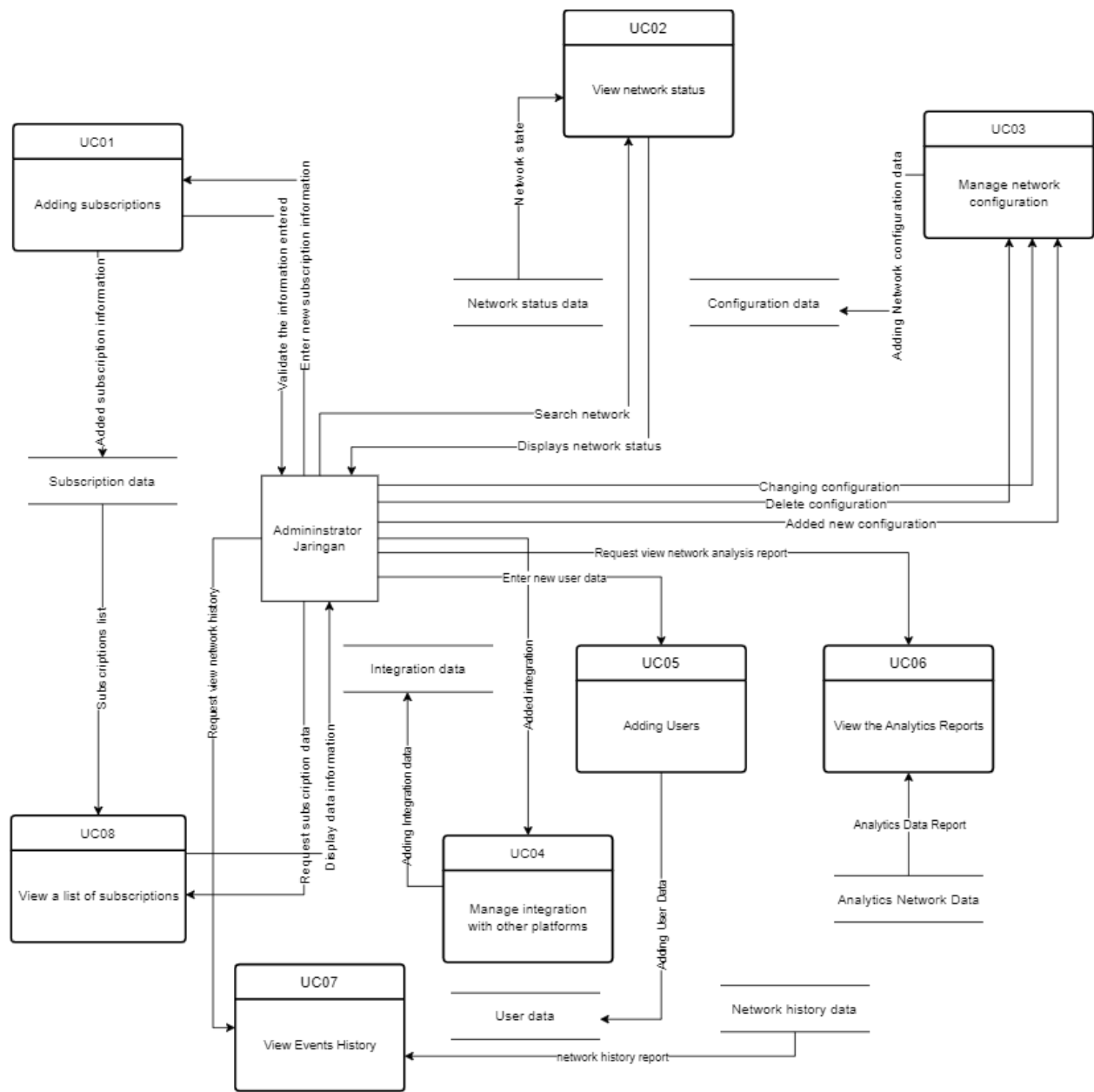
- Acceptance Criteria:

1. Saya harus dapat mengakses daftar subscription yang sudah ditambahkan di dashboard Netmonk.
2. Sistem harus menampilkan informasi detail untuk setiap subscription, seperti nama subscription, alamat IP jaringan yang dimonitor, dan credential yang diperlukan.
3. Sistem harus menampilkan status subscription, apakah subscription aktif atau tidak aktif.

Soal no 4

a. Data Flow Diagram

1. Level 0



b. Use Case Diagram



c. Activity Diagram

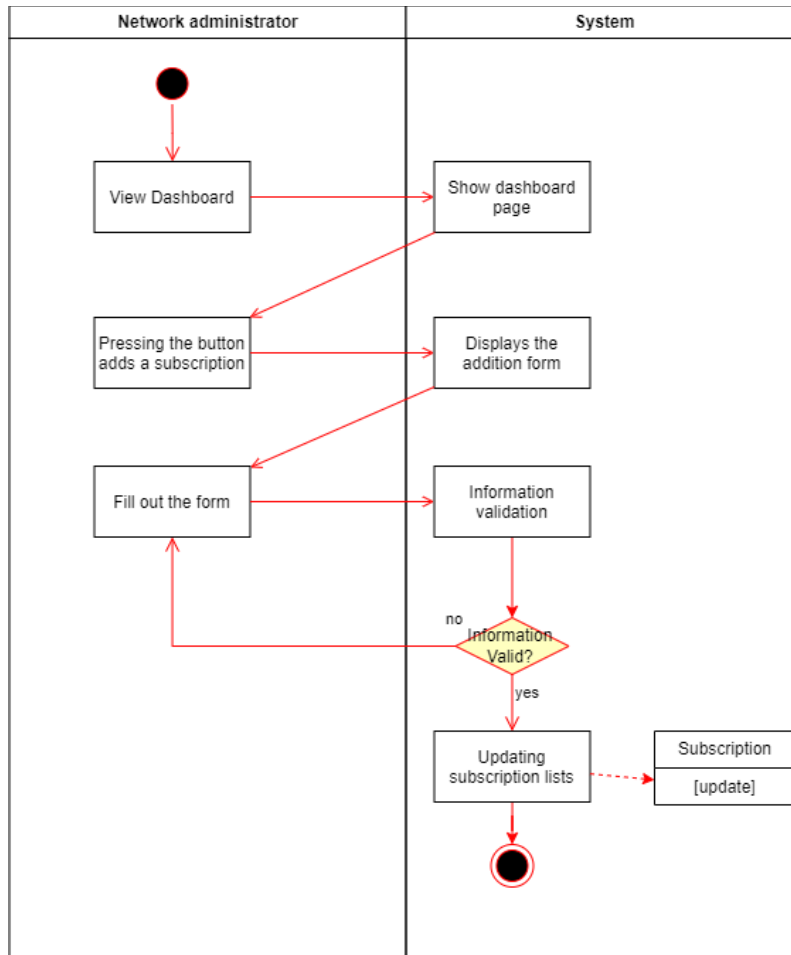
1. Use Case Specification

Use Case Specification

1. UC01

Use Case Code	UC01
Use Case Name	Adding Subscription
Actor	Network administrator
Description	This use case describes how a network administrator adds a subscription or subscription service to gain access to features in Netmonk.
Relation	

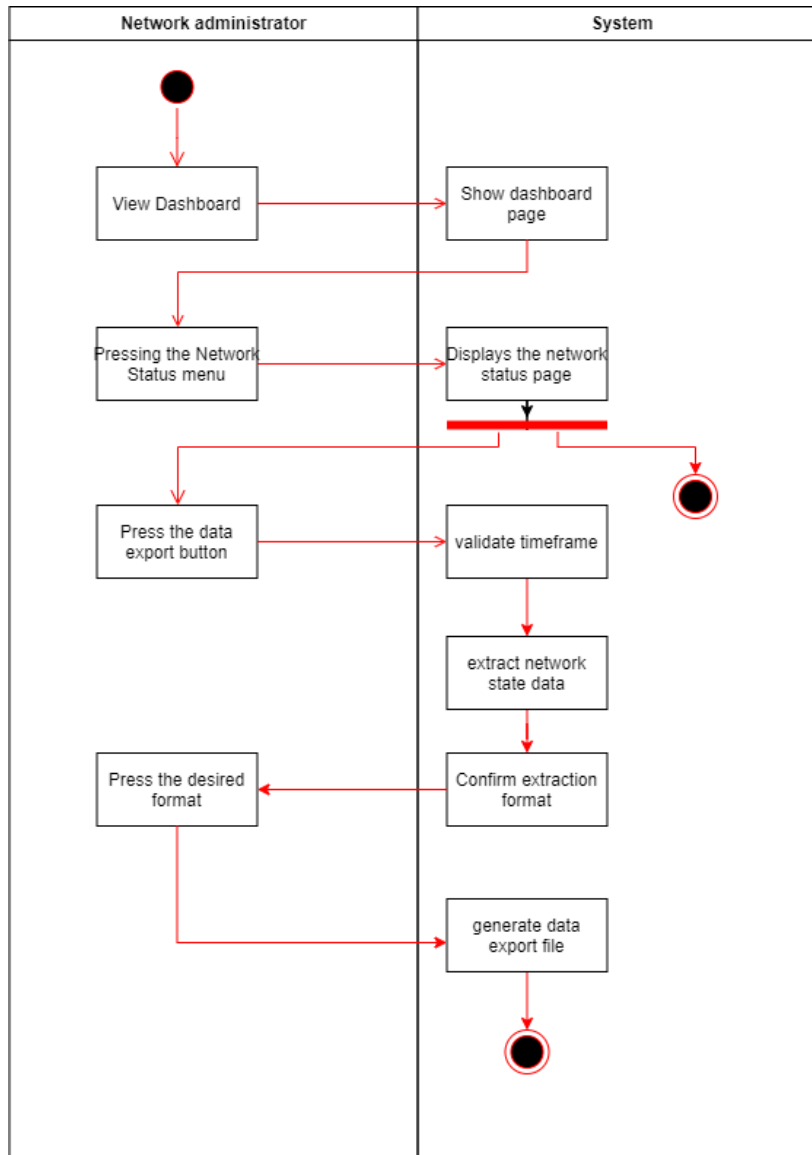
Initial conditions	No additional subscriptions	
Final condition	There is an additional subscription	
Normal Flow		
	Network administrator	System
	1. Open the Netmonk dashboard	1.1 Displays the dashboard page
	2. Select the add subscription button	2.1 Displays the form to add a subscription
	3. Enter the required subscription information	3.1.a Validate the information entered
	4. View the list of subscriptions that have been added	4.1 Displays a list of subscriptions that have been completed
Alternative Flow		
	3.1.a	
	1. The network administrator enters incomplete data 2. The system again asks to complete the form 3. Administrator completes the form 4. Back to normal flow no 4.	



2. UC02

Use Case Code	UC02	
Use Case Name	View Network Status	
Actor	Network administrator	
Description	This use case describes how network administrators view network status on the Netmonk dashboard.	
Relation		
Initial conditions	Administrator berada dalam halaman dashboard Netmonk	
Final condition	Administrator berada dalam halaman status jaringan	
Normal Flow		
	Network administrator	System

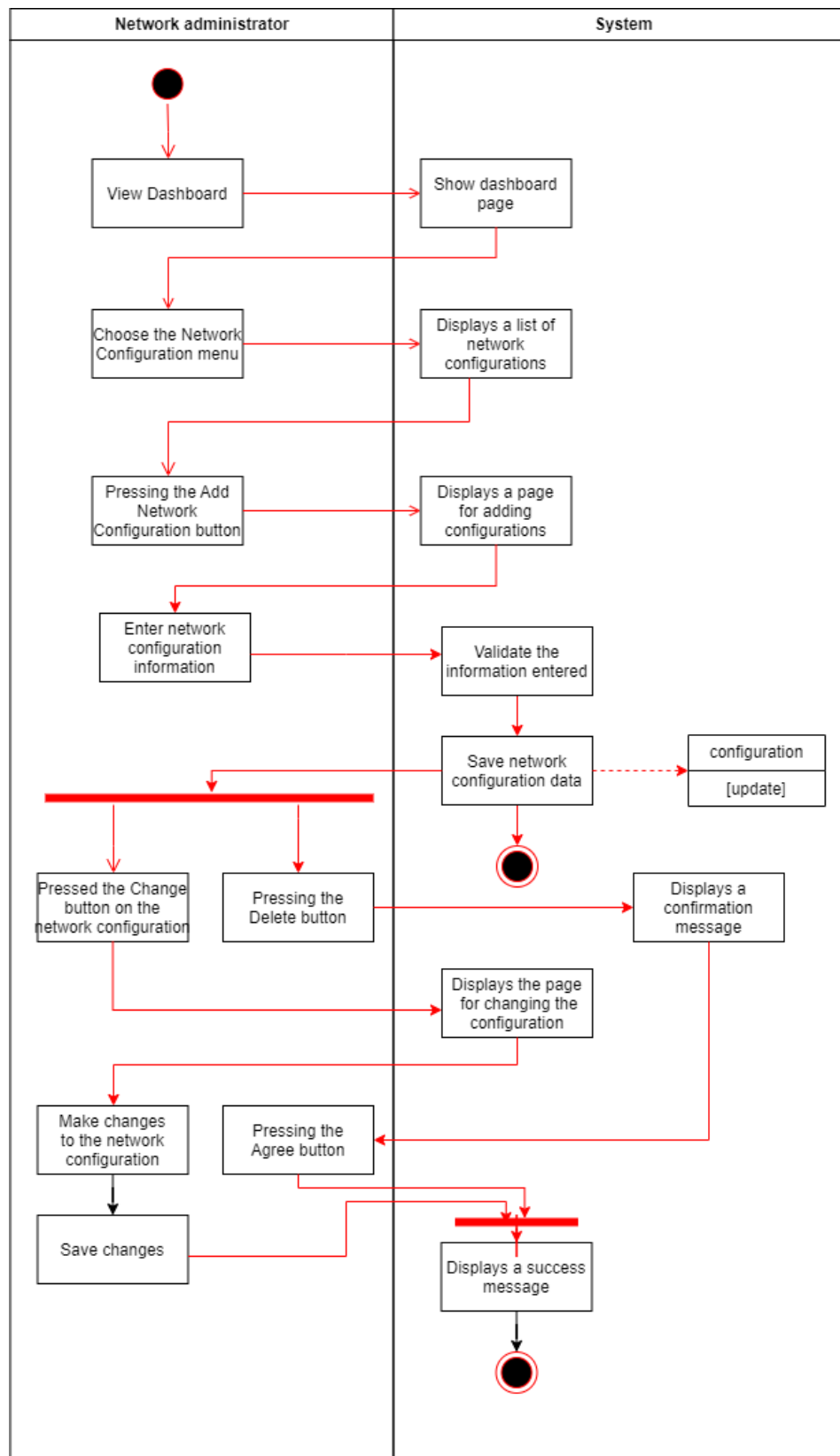
	1. Membuka halaman dashboard	1.1 Menampilkan halaman dashboard
	2. Menekan menu Status Jaringan	2.1 Menampilkan halaman status jaringan
Alternative Flow		
Extension		
	2.1.a Administrator jaringan Melakukan Ekspor data status jaringan 2.1.a.1 Administrator jaringan menekan tombol Ekspor Data 2.1.a.2.a Sistem memvalidasi rentang waktu 2.1.a.2.b Sistem mengekstrak data status jaringan 2.1.a.2.c Sistem menampilkan halaman konfirmasi format ekstraksi untuk pengguna 2.1.a.3 Pengguna menekan format file yang diinginkan untuk melakukan konfirmasi 2.1.a.4 Sistem menghasilkan file ekspor data	



3. UC03

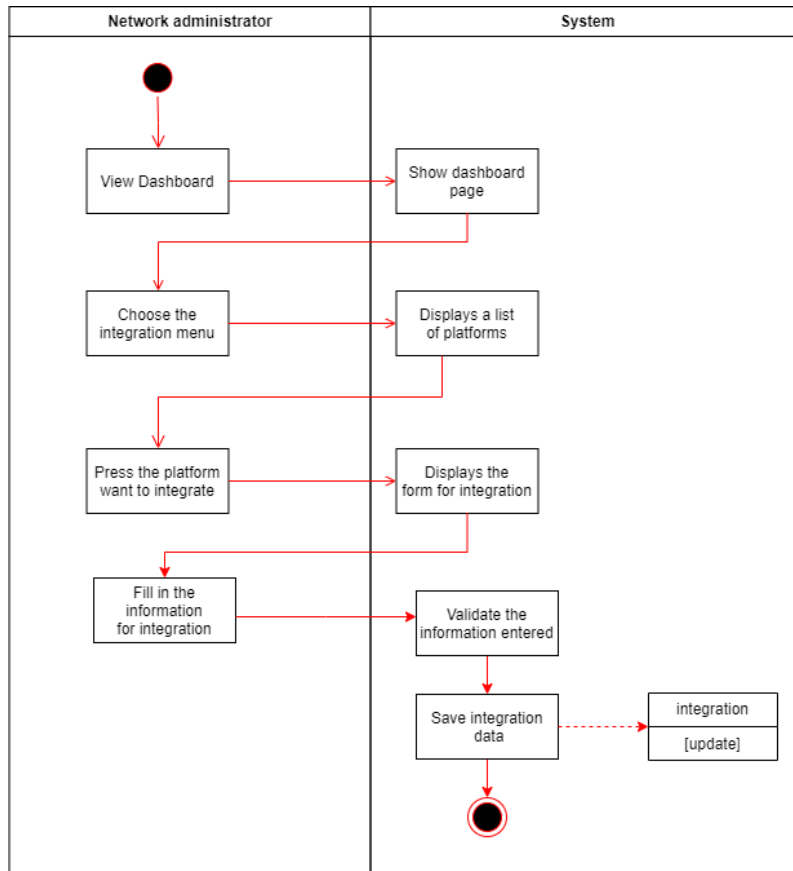
Use Case Code	UC03
Use Case Name	Manage Network Configuration
Actor	Network administrator
Description	This use case describes how a network administrator manages network configuration using the features available on Netmonk.
Relation	

Initial conditions	Administrator jaringan berada dalam dashboard Netmonk	
Final condition	Administrator jaringan melakukan perubahan pada konfigurasi jaringan	
Normal Flow		
	Network administrator	System
	1 Memilih menu Konfigurasi Jaringan	1.1 Menampilkan daftar konfigurasi jaringan yang telah ditambahkan sebelumnya
	2 Menambahkan konfigurasi jaringan baru dengan menekan tombol Tambah Konfigurasi Jaringan	2.1 Menampilkan halaman untuk menambahkan konfigurasi jaringan baru
	3 Memasukkan informasi konfigurasi jaringan	3.1 Memvalidasi informasi yang dimasukkan pengguna 3.2 Menyimpan konfigurasi jaringan dalam database 3.3 Menampilkan pesan sukses
Alternative Flow		
Extension		
	<p>1.a Administrator jaringan dapat Mengubah Konfigurasi jaringan yang telah ditambahkan sebelumnya</p> <p>1.a.1 Pengguna menekan tombol Ubah pada konfigurasi jaringan yang ingin dirubah</p> <p>1.a.2 Sistem menampilkan halaman untuk merubah konfigurasi jaringan yang dipilih</p> <p>1.a.3.a Pengguna melakukan perubahan pada konfigurasi jaringan yang dipilih</p> <p>1.a.3.b Pengguna menyimpan perubahan dengan menekan tombol Simpan Perubahan</p> <p>1.a.4.a Sistem memvalidasi perubahan informasi</p> <p>1.a.4.b Sistem menampilkan pesan sukses</p> <p>1.b Administrator jaringan dapat Menghapus Konfigurasi jaringan yang telah ditambahkan sebelumnya</p> <p>1.b.1 Pengguna menekan tombol Hapus pada daftar konfigurasi jaringan</p> <p>1.b.2 Sistem menampilkan pesan konfirmasi hapus</p> <p>1.b.3 Pengguna menekan tombol Setuju</p> <p>1.b.4 Sistem menghapus konfigurasi jaringan</p>	



4. UC04

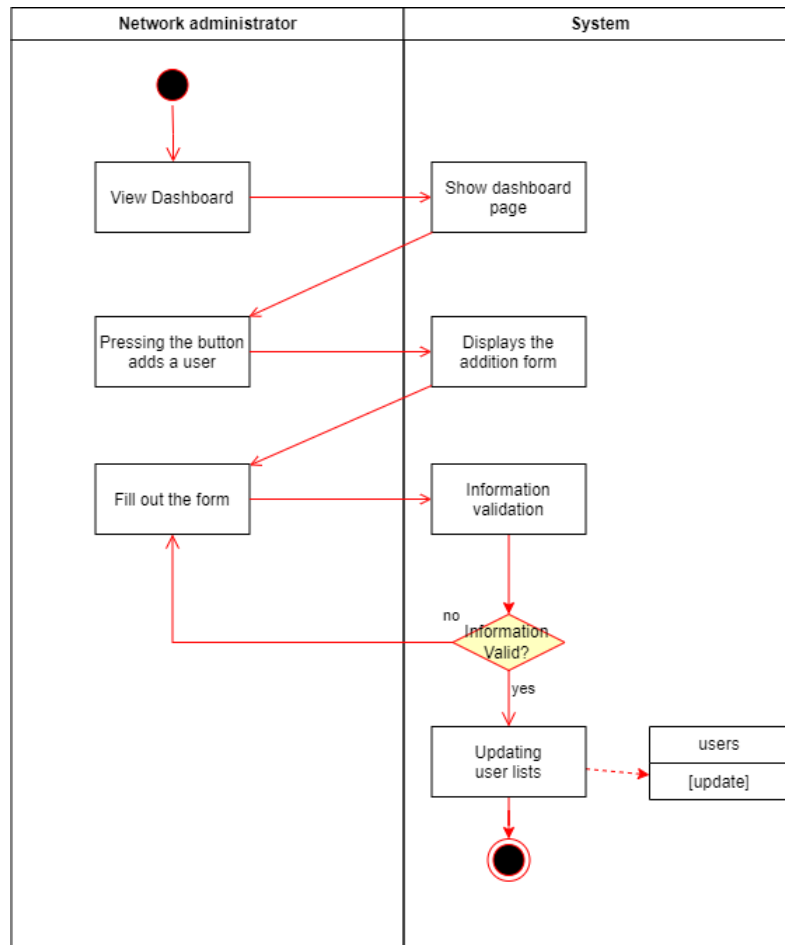
Use Case Code	UC04	
Use Case Name	Managing Integration with Other Platforms	
Actor	Network administrator	
Description	This use case describes how network administrators manage integration with other platforms by using the features available on Netmonk.	
Relation		
Initial conditions	Administrator jaringan berada pada halaman dashboard Netmonk	
Final condition	Terjadi penambahan integrasi pada database	
Normal Flow		
	Network administrator	System
	1 Menekan menu Integrasi pada navigasi	1.1 Menampilkan daftar platform yang telah terintegrasi
	2 Menekan tombol Tambah Integrasi	2.1 Menampilkan daftar platform yang dapat diintegrasikan dengan netmonk
	3 Menekan platform yang ingin diintegrasikan	3.1 Menampilkan form untuk integrasi
	4 Mengisi informasi untuk integrasi	4.1 Memvalidasi informasi yang dimasukkan
Alternative Flow		



5. UC05

Use Case Code	UC05	
Use Case Name	Add User	
Actor	Network administrator	
Description	This use case describes how network administrators view the reports provided by Netmonk.	
Relation		
Initial conditions	Administrator jaringan berada pada halaman dashboard	
Final condition	Administrator jaringan berhasil menambahkan pengguna baru	
Normal Flow		
	Network administrator	System
	1 Menekan menu Tambah Pengguna Baru	1.1 Menampilkan form untuk memasukkan informasi pengguna

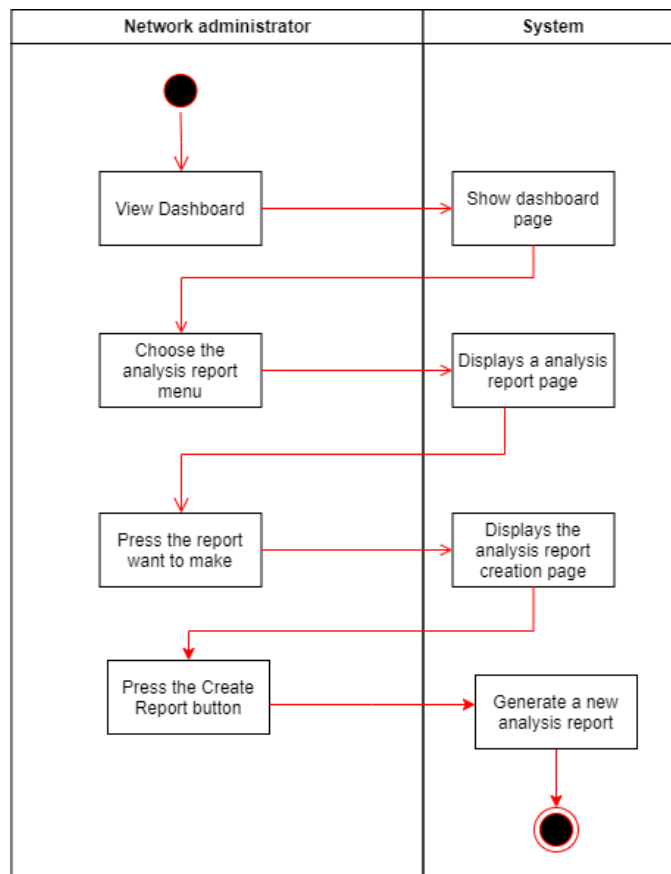
	2 Memasukkan informasi pengguna baru	2.1 Sistem memvalidasi informasi yang dimasukkan
Alternative Flow		



6. UC06

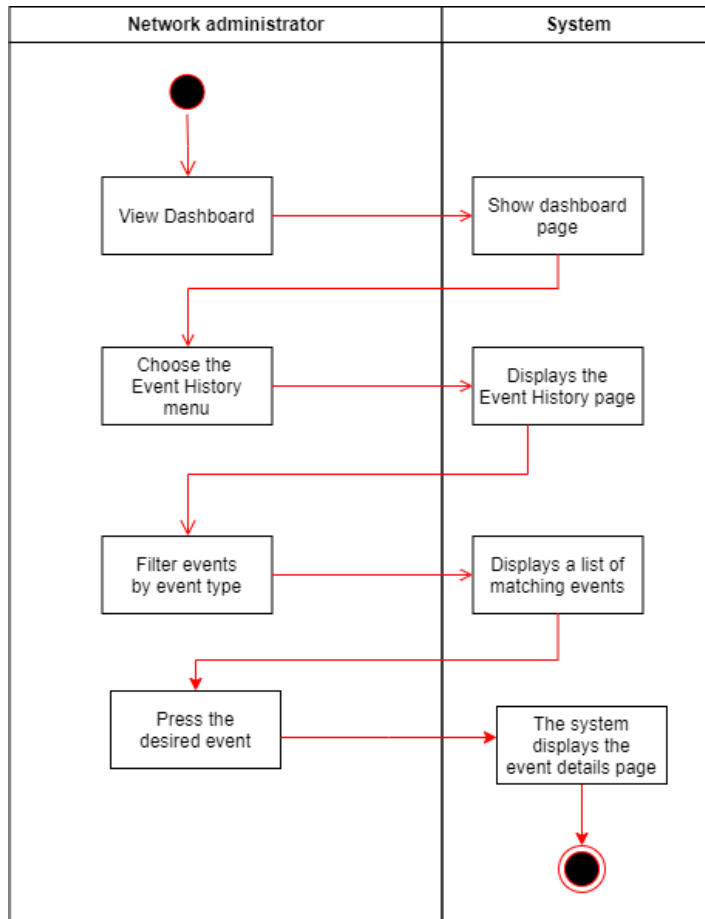
Use Case Code	UC06
Use Case Name	View the Analytics Report
Actor	Network administrator
Description	This use case describes how network administrators view the reports provided by Netmonk.
Relation	

Initial conditions	Administrator jaringan berada dalam halaman dashboard	
Final condition	Administrator jaringan berhasil melihat laporan analisis	
Normal Flow		
	Network administrator	System
	1 Menekan menu Laporan Analisis	1.1 Menampilkan halaman laporan analisis
	2 Menekan salah satu laporan analisis yang ada	2.1 Menampilkan halaman pembuatan laporan analisis
	3 Menekan tombol Buat Laporan	3.1 Menghasilkan laporan analisis baru
Alternative Flow		



7. UC07

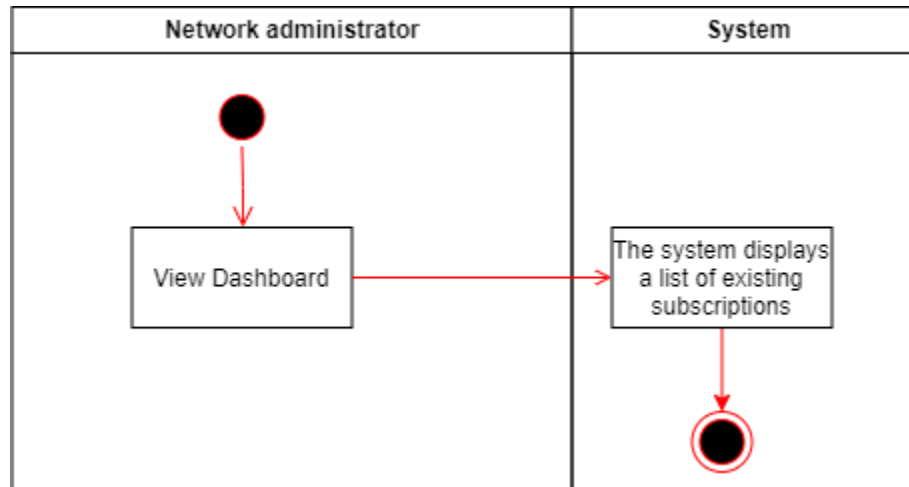
Use Case Code	UC07	
Use Case Name	View history of events	
Actor	Network administrator	
Description	This use case describes how network administrators view the event history and network usage provided by Netmonk.	
Relation		
Initial conditions	Administrator jaringan berada pada halaman dashboard	
Final condition	Administrator jaringan berhasil melihat riwayat peristiwa pada Netmonk	
Normal Flow		
	Network administrator	System
	1 Menekan menu Riwayat Peristiwa	1.1 Menampilkan halaman Riwayat Peristiwa
	2 Memfilter peristiwa berdasarkan jenis peristiwa	2.1 Menampilkan daftar peristiwa yang sesuai
	3 Menekan peristiwa yang diinginkan	3.1 Sistem menampilkan halaman rincian peristiwa
Alternative Flow		



8. UC08

Use Case Code	UC08	
Use Case Name	View a list of subscriptions	
Actor	Network administrator	
Description	This use case describes how a network administrator views the list of subscriptions that have been added	
Relation		
Initial conditions	Administrator berada pada halaman dashboard	
Final condition	Administrator jaringan berhasil melihat daftar subscription	
Normal Flow		
	Network administrator	System

	1 Membuka dashboard	1.1 Sistem menampilkan daftar subscription yang ada.
Alternative Flow		



Soal no 5

Ini penjelasan (rancangan) untuk API no 5

1. API untuk menambahkan subscription

Deskripsi:

API ini akan memungkinkan pengguna untuk menambahkan subscription baru ke dalam sistem. Setelah pengguna memasukkan informasi subscription, API akan memvalidasi informasi tersebut dan menyimpannya di dalam database.

Rincian API:

- Endpoint: POST /subscriptions

- Input parameter:

- subscription_name (string, wajib): Nama subscription yang akan ditambahkan.
- subscription_type (string, wajib): Jenis subscription, seperti Premium atau Basic.
- subscription_period (string, wajib): Periode berlangganan, misalnya bulanan atau tahunan.
- subscription_price (number, wajib): Harga berlangganan.

- Output:

- status_code (number): Kode status HTTP, misalnya 200 OK atau 400 Bad Request.
- message (string): Pesan sukses atau gagal.
- data (object): Objek subscription baru yang telah ditambahkan.

2. API untuk mengelola konfigurasi jaringan

Deskripsi:

API ini akan memungkinkan pengguna untuk menambahkan, mengubah, atau menghapus perangkat dan topologi jaringan serta menyimpan informasi konfigurasi jaringan di dalam database.

Rincian API:

- Endpoint:

- POST /devices: Menambahkan perangkat baru ke dalam sistem.
- PUT /devices/{device_id}: Mengubah informasi perangkat yang sudah ada.
- DELETE /devices/{device_id}: Menghapus perangkat yang sudah ada.
- POST /topologies: Menambahkan topologi baru ke dalam sistem.
- PUT /topologies/{topology_id}: Mengubah informasi topologi yang sudah ada.
- DELETE /topologies/{topology_id}: Menghapus topologi yang sudah ada.

- Input parameter:

- Device:

- device_name (string, wajib): Nama perangkat.
- device_ip (string, wajib): Alamat IP perangkat.
- device_type (string, wajib): Jenis perangkat, misalnya router atau switch.
- device_location (string, wajib): Lokasi perangkat.
- device_status (string, opsional): Status perangkat, misalnya online atau offline.

- Topology:

- topology_name (string, wajib): Nama topologi.
- topology_description (string, opsional): Deskripsi topologi.
- devices (array of object, wajib): Array berisi objek-objek perangkat yang terkait dengan topologi

tersebut.

- Output:

- status_code (number): Kode status HTTP, misalnya 200 OK atau 400 Bad Request.
- message (string): Pesan sukses atau gagal.
- data (object): Objek perangkat atau topologi yang telah ditambahkan, diubah, atau dihapus.

Tentu, berikut adalah deskripsi dan rincian API untuk use case yang telah dijelaskan sebelumnya:

3. API Integrasi dengan Platform Lain

Deskripsi: API ini memungkinkan pengguna untuk mengintegrasikan Netmonk dengan platform lain seperti Slack, Microsoft Teams, atau layanan help desk lainnya. API ini juga memvalidasi informasi integrasi dengan platform lain dan menyimpan informasi integrasi dengan platform lain di database.

- Endpoint: /integrations

- Metode: POST

- Input:

- platform_name: string (required)
- platform_api_key: string (required)

- platform_api_secret: string (optional)
- Output:
 - Status code 200 jika integrasi berhasil ditambahkan
 - Status code 400 jika input tidak valid
 - Status code 401 jika pengguna tidak memiliki akses ke fitur ini
 - Status code 500 jika terjadi kesalahan internal server

4. API Menambahkan Pengguna Baru

Deskripsi: API ini memungkinkan Administrator Jaringan untuk menambahkan pengguna baru, memvalidasi informasi pengguna, dan menyimpan informasi pengguna di database.

- Endpoint: /users
- Metode: POST
- Input:
 - name: string (required)
 - email: string (required)
 - password: string (required)
 - role: string (required, value: admin or user)
- Output:
 - Status code 200 jika pengguna berhasil ditambahkan
 - Status code 400 jika input tidak valid
 - Status code 401 jika pengguna tidak memiliki akses ke fitur ini
 - Status code 500 jika terjadi kesalahan internal server

5. API Melihat Riwayat Peristiwa Jaringan

Deskripsi: API ini memungkinkan Administrator Jaringan untuk melihat riwayat peristiwa jaringan, termasuk pemberitahuan masalah dan tindakan yang diambil. API ini juga memungkinkan Administrator Jaringan untuk memilih rentang waktu riwayat peristiwa dan menampilkan riwayat peristiwa dalam format yang mudah dibaca.

- Endpoint: /network_events
- Metode: GET
- Input:
 - start_time: datetime (optional)
 - end_time: datetime (optional)
- Output:
 - Data riwayat peristiwa dalam format yang mudah dibaca, seperti JSON atau CSV.
 - Status code 200 jika riwayat peristiwa berhasil diambil
 - Status code 401 jika pengguna tidak memiliki akses ke fitur ini
 - Status code 500 jika terjadi kesalahan internal server

6. API untuk melihat laporan analisis jaringan

- Deskripsi: API ini memungkinkan Administrator Jaringan untuk melihat laporan analisis jaringan pada rentang waktu tertentu dan menampilkan laporan dalam format yang mudah dibaca.
- Endpoint: /api/network/report
- Metode: GET
- Parameter:
 - start_date (wajib): tanggal awal rentang waktu yang ingin dilihat (format: yyyy-mm-dd)
 - end_date (wajib): tanggal akhir rentang waktu yang ingin dilihat (format: yyyy-mm-dd)
- Response:
 - status: status kesuksesan operasi (bernilai true jika berhasil dan false jika gagal)
 - message: pesan penjelasan status
 - report: data laporan analisis jaringan dalam format yang mudah dibaca

7. API untuk melihat status jaringan secara real-time

- Deskripsi: API ini memungkinkan Administrator Jaringan untuk melihat status jaringan secara real-time, termasuk ketersediaan perangkat jaringan dan kinerja jaringan dan menampilkan status jaringan dalam format yang mudah dibaca.
- Endpoint: /api/network/status
- Metode: GET
- Parameter: tidak ada parameter tambahan yang dibutuhkan
- Response:
 - status: status kesuksesan operasi (bernilai true jika berhasil dan false jika gagal)
 - message: pesan penjelasan status
 - status_data: data status jaringan dalam format yang mudah dibaca

8. Memungkinkan Administrator Jaringan untuk melihat laporan analisis jaringan, termasuk pemantauan kinerja dan keamanan jaringan

Endpoint: GET /network-analysis

Deskripsi: Mengambil laporan analisis jaringan dalam rentang waktu tertentu. Pengguna dapat memilih rentang waktu dan jenis analisis yang ingin dilihat.

Parameter:

- start_time (opsional): waktu mulai rentang waktu laporan analisis yang ingin dilihat. Format: timestamp ISO 8601 (YYYY-MM-DDTHH:MM:SSZ).
- end_time (opsional): waktu akhir rentang waktu laporan analisis yang ingin dilihat. Format: timestamp ISO 8601 (YYYY-MM-DDTHH:MM:SSZ).
- analysis_type (opsional): jenis analisis yang ingin dilihat (contoh: performace, security, dll).

Response:

- status (integer): kode status HTTP yang menunjukkan berhasil atau gagalnya request.
- message (string): pesan yang menjelaskan status request.

- analysis (array of objects): array yang berisi informasi laporan analisis jaringan. Setiap objek berisi informasi seperti jenis analisis, hasil analisis, dan waktu analisis.

9. Memungkinkan Administrator Jaringan untuk melihat status jaringan secara real-time, termasuk ketersediaan perangkat jaringan dan kinerja jaringan dan menampilkan status jaringan dalam format yang mudah dibaca.

Endpoint: GET /network-status

Deskripsi: Mengambil status jaringan secara real-time. API ini akan menampilkan informasi tentang ketersediaan perangkat jaringan dan kinerja jaringan dalam format yang mudah dibaca.

Parameter: tidak ada.

Response:

- status (integer): kode status HTTP yang menunjukkan berhasil atau gagalnya request.
- message (string): pesan yang menjelaskan status request.
- network_status (object): objek yang berisi informasi tentang status jaringan. Objek ini akan mencakup informasi seperti ketersediaan perangkat, status koneksi, dan kinerja jaringan.

Tabel dan relasinya adalah sebagai berikut:

Tabel subscription:

- subs_id (int) primary key
- subs_name (varchar 100)
- subs_period (date)
- subs_price
- subs_status

Tabel user:

- user_id (int 10) primary key
- user_name (varchar 50)
- user_email (varchar 50)
- user_password (varchar 50)
- user_position
- user_department
- user_account_status
- user_role

Tabel device:

- dev_id primary key
- dev_name
- dev_type

- dev_status
- dev_location
- dev_ip

Tabel connection:

- conn_id
- conn_type
- conn_status
- dev_id (foreign key ke tabel device)

Tabel network_interface:

- net_interface_id
- net_interface_name
- net_interface_type
- net_interface_status
- dev_id (foreign key ke tabel device)

Tabel topology:

- topology_id primary key
- topology_name
- topology_desc

Tabel topology_device:

- topology_device_id primary key
- topology_id (foreign key ke tabel topology)
- dev_id (foreign key ke tabel device)

Tabel event:

- event_id primary key
- event_type
- event_time
- event_desc
- dev_id (foreign key ke tabel device)

Tabel report:

- report_id primary key
- report_start_date
- report_end_date
- report_data
- dev_id (foreign key ke tabel device)

Tabel platform:

- platform_id primary key
- platform_name

- platform_api_key
- platform_api_secret

Tabel network_analysis:

- network_analysis_id primary key
- network_analysis_start_time
- network_analysis_end_time
- network_analysis_type
- dev_id (foreign key ke tabel device)

Relasi antar tabel:

- Tabel subscription berelasi dengan tabel user (many-to-one, foreign key pada tabel user).
- Tabel device berelasi dengan tabel connection (one-to-many, foreign key pada tabel connection).
- Tabel device berelasi dengan tabel network_interface (one-to-many, foreign key pada tabel network_interface).
- Tabel device berelasi dengan tabel event (one-to-many, foreign key pada tabel event).
- Tabel device berelasi dengan tabel report (one-to-many, foreign key pada tabel report).
- Tabel topology_device berelasi dengan tabel topology (many-to-one, foreign key pada tabel topology).
- Tabel connection berelasi dengan tabel device (many-to-one, foreign key pada tabel device).
- Tabel network_interface berelasi dengan tabel device (many-to-one, foreign key pada tabel device).
- Tabel event berelasi dengan tabel device (many-to-one, foreign key pada tabel device).
- Tabel report berelasi dengan tabel device (many-to-one, foreign key pada tabel device).
- Tabel network_analysis berelasi dengan tabel device (many-to-one, foreign key pada tabel device).

Soal no 6

1. GET /api/status (O(1))

- API ini digunakan untuk mendapatkan status jaringan dari sistem monitoring.

2. POST /api/subscriptions (O(n))

- API ini digunakan untuk menambahkan subscription baru ke sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data subscription yang sudah ada di dalam database.

3. GET /api/subscriptions (O(n))

- API ini digunakan untuk mendapatkan daftar subscription yang sudah ada di dalam sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data subscription yang sudah ada di dalam database.

4. DELETE /api/subscriptions/{id} (O(1))

- API ini digunakan untuk menghapus subscription dari sistem monitoring.

5. GET /api/reports (O(n))

- API ini digunakan untuk mendapatkan laporan analisis yang sudah dilakukan. Kompleksitas Big O-nya bergantung pada jumlah data laporan yang sudah ada di dalam database.

6. POST /api/users ($O(n)$)

- API ini digunakan untuk menambahkan pengguna baru ke sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data pengguna yang sudah ada di dalam database.

7. GET /api/users ($O(n)$)

- API ini digunakan untuk mendapatkan daftar pengguna yang sudah terdaftar di dalam sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data pengguna yang sudah ada di dalam database.

8. DELETE /api/users/{id} ($O(1)$)

- API ini digunakan untuk menghapus pengguna dari sistem monitoring.

9. GET /api/history ($O(n)$)

- API ini digunakan untuk mendapatkan riwayat peristiwa yang terjadi di dalam sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data riwayat peristiwa yang sudah ada di dalam database.

10. GET /api/integrations ($O(n)$)

- API ini digunakan untuk mendapatkan daftar integrasi dengan platform lain yang sudah terdaftar di dalam sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data integrasi yang sudah ada di dalam database.

11. POST /api/integrations ($O(n)$)

- API ini digunakan untuk menambahkan integrasi dengan platform lain ke sistem monitoring. Kompleksitas Big O-nya bergantung pada jumlah data integrasi yang sudah ada di dalam database.

12. DELETE /api/integrations/{id} ($O(1)$)

- API ini digunakan untuk menghapus integrasi dengan platform lain dari sistem monitoring.

13. GET /api/configuration ($O(1)$)

- API ini digunakan untuk mendapatkan konfigurasi jaringan yang sudah diatur di dalam sistem monitoring.

14. PUT /api/configuration ($O(1)$)

- API ini digunakan untuk mengubah konfigurasi jaringan di dalam sistem monitoring.