



POLITEKNIK STATISTIKA STIS

Rancangan Sistem dan Program

Pengembangan Aplikasi Website 'Sirema'
Sistem *Request* Media Kampus

Disusun Oleh:

Tim 3 – 3S11



TIM PENYUSUN

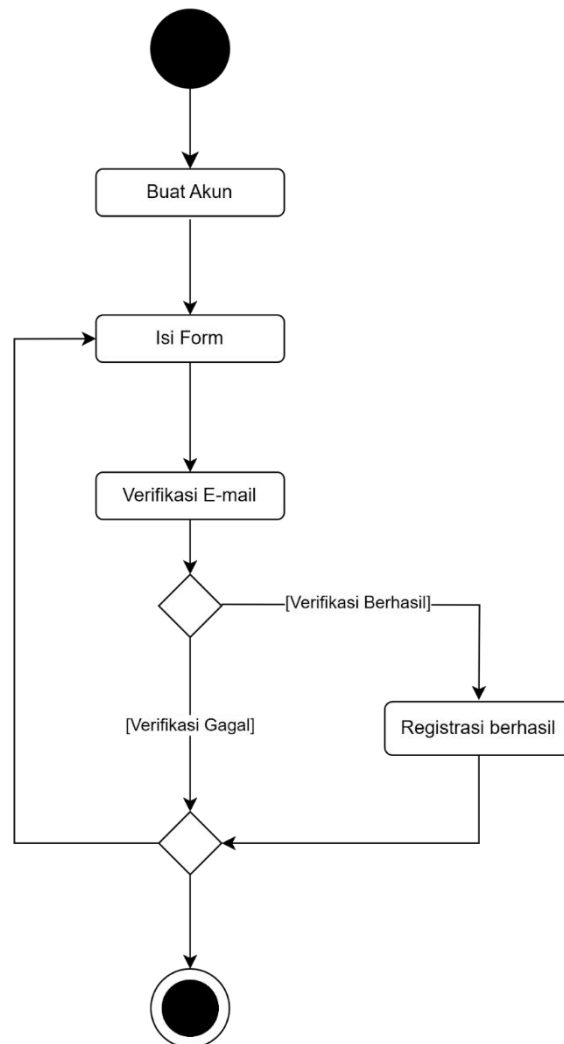
Nama	NIM
Aditya Widiyanto Nugroho	222111845
Akma Batrisyia Jazima	222111871
I Made Yoga Andika Putra	222112102
Irsyad Fadhil Asyraf	222112116
Jihan Maisaroh	222112122
Luthfiani Nur Aisyah	222112162
Samuel Maruba Manik	222112348
Yoga Pratama	222112419

DAFTAR ISI

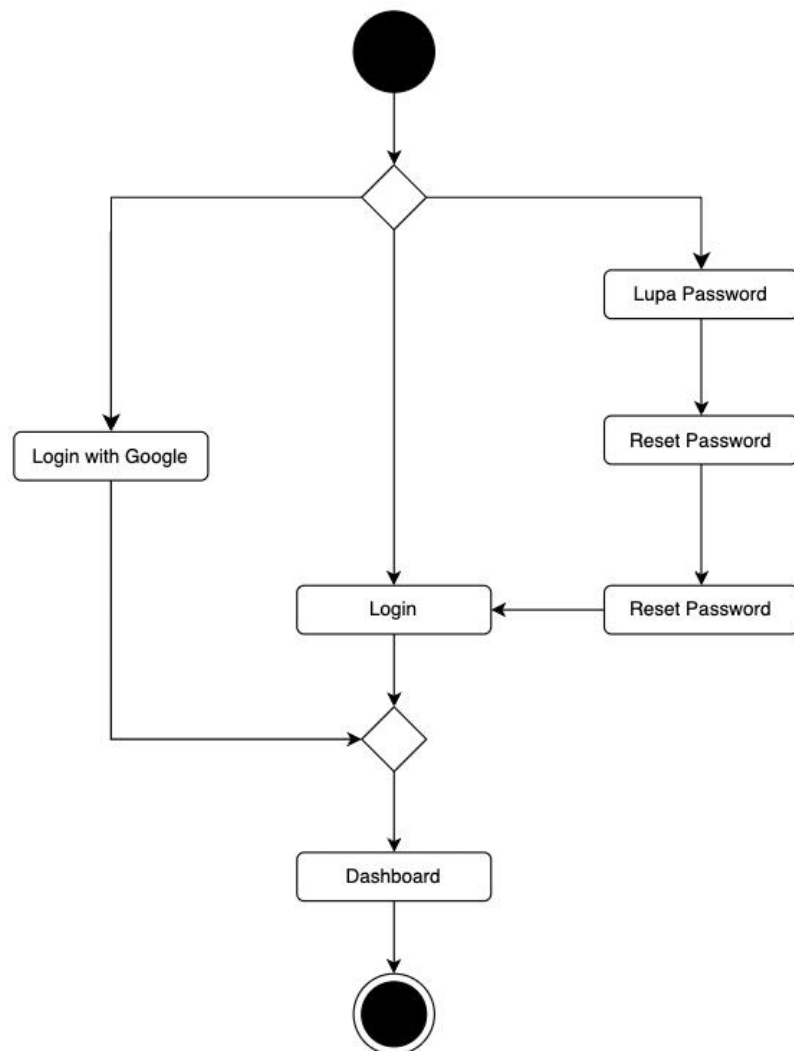
DAFTAR ISI.....	ii
1. Flowchart/ Activity Diagram	1
1.1 Registrasi (User)	1
1.2 Login (Admin, User, Personel)	2
1.3 Manajemen Akun (Admin)	3
1.4 Request (Admin, User)	4
1.5 Penerimaan/ Penolakan Request (Admin, Personel)	5
2. Diagram Use Case	7
2.1 Registrasi (User)	7
2.2 Login (Admin, User, Personel)	7
2.3 Manajemen Akun (Admin)	8
2.4 Request (Admin, User)	8
2.5 Penerimaan/ Penolakan Request (Admin, Personel)	9
3. Class Diagram	10
4. Sequence Diagram	12
5. Rancangan Basis Data	17
6. Rancangan Arsitektur Sistem	20
7. Design Pattern	22

1. Flowchart/ Activity Diagram

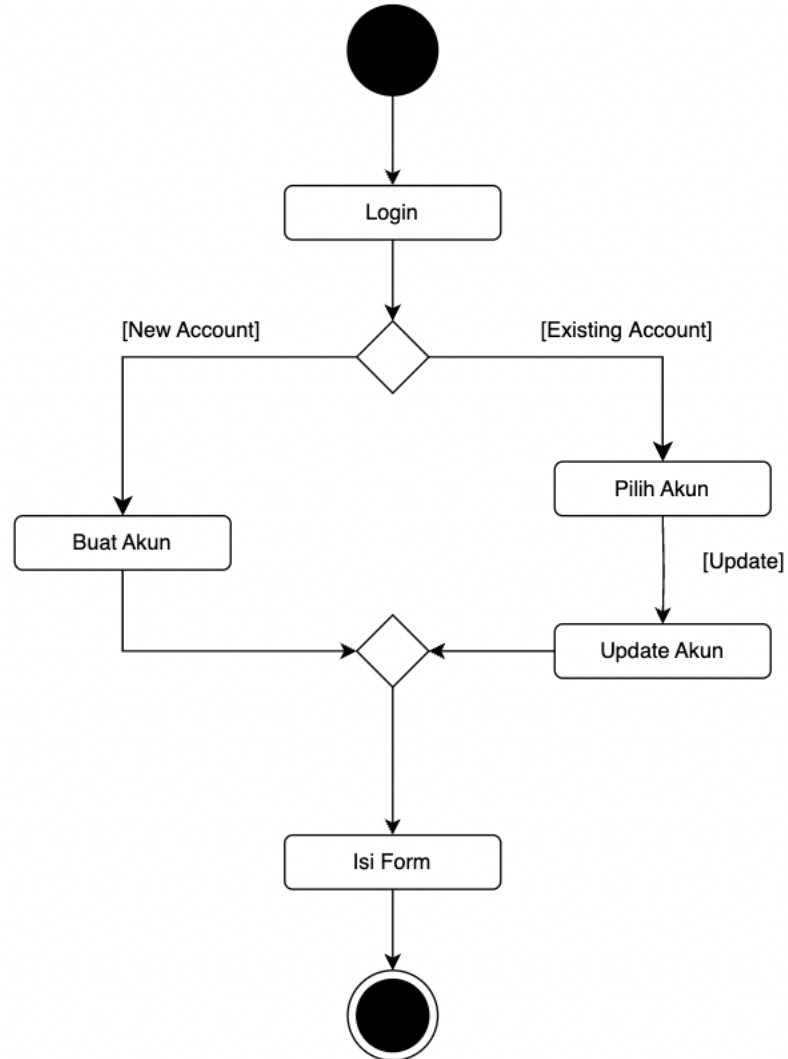
1.1 Registrasi (User)



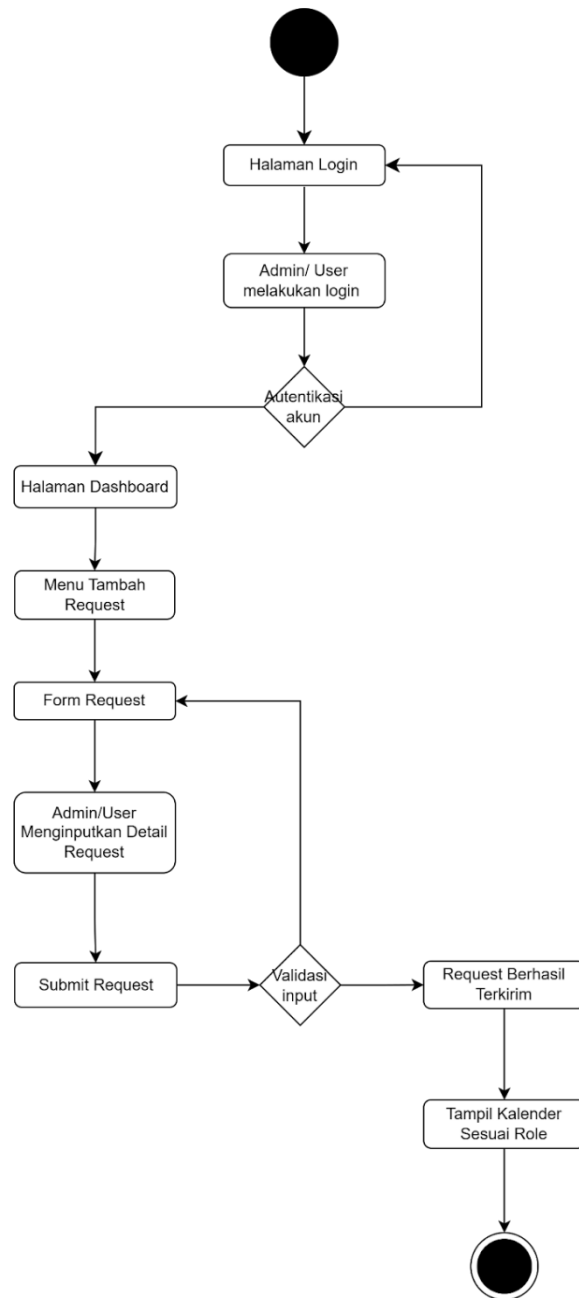
1.2 Login (Admin, User, Personel)



1.3 Manajemen Akun (Admin)

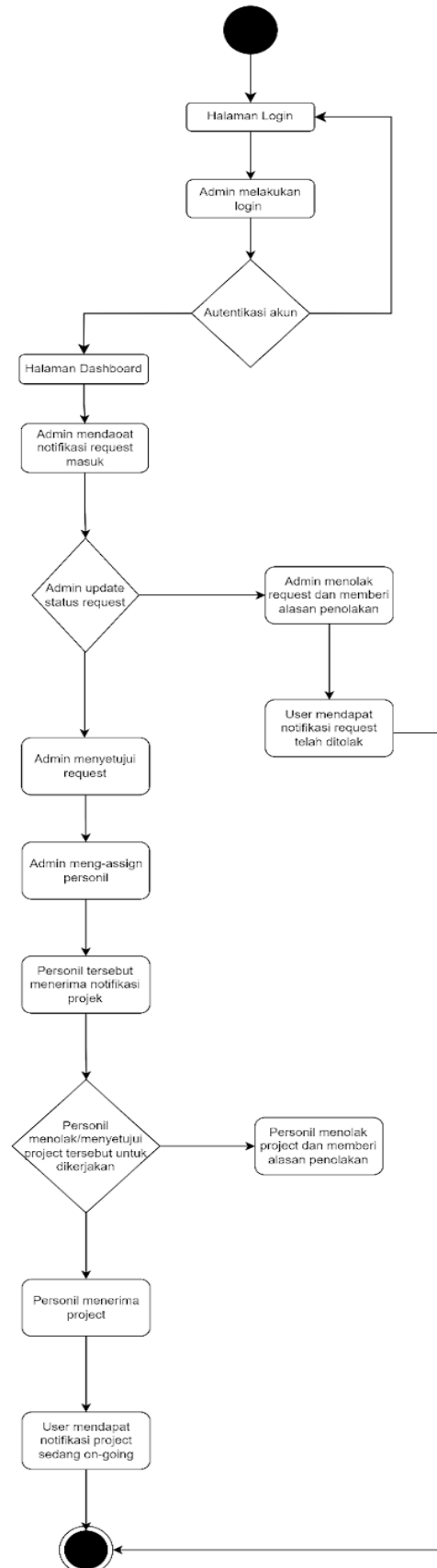


1.4 Request (Admin, User)



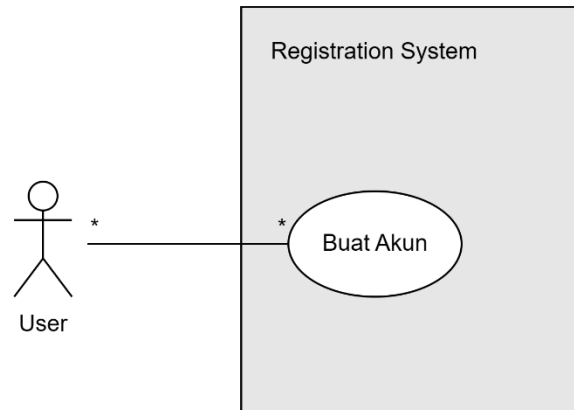


1.5 Penerimaan/Penolakan Request (Admin, Personel)

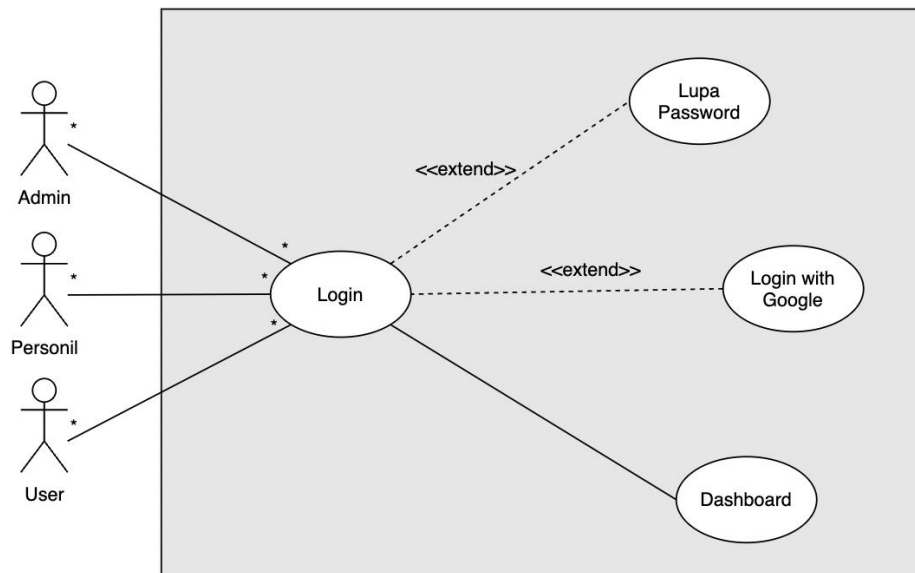


2. Diagram Use Case

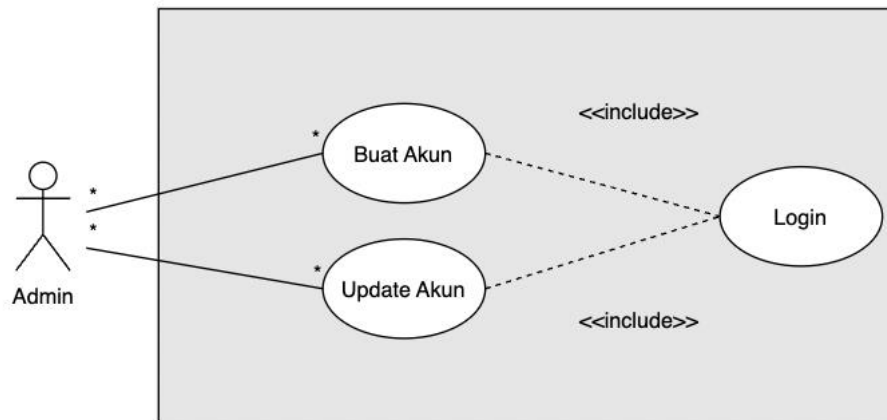
2.1 Registrasi (User)



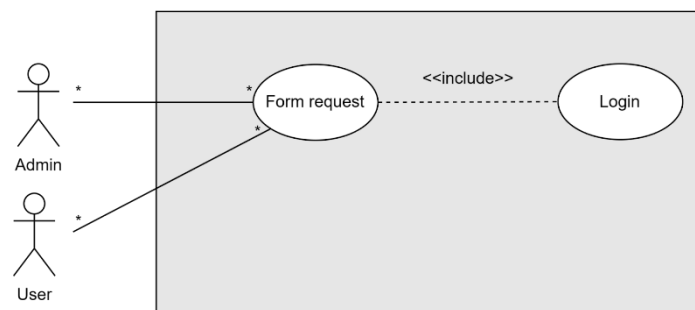
2.2 Login (Admin, User, Personel)



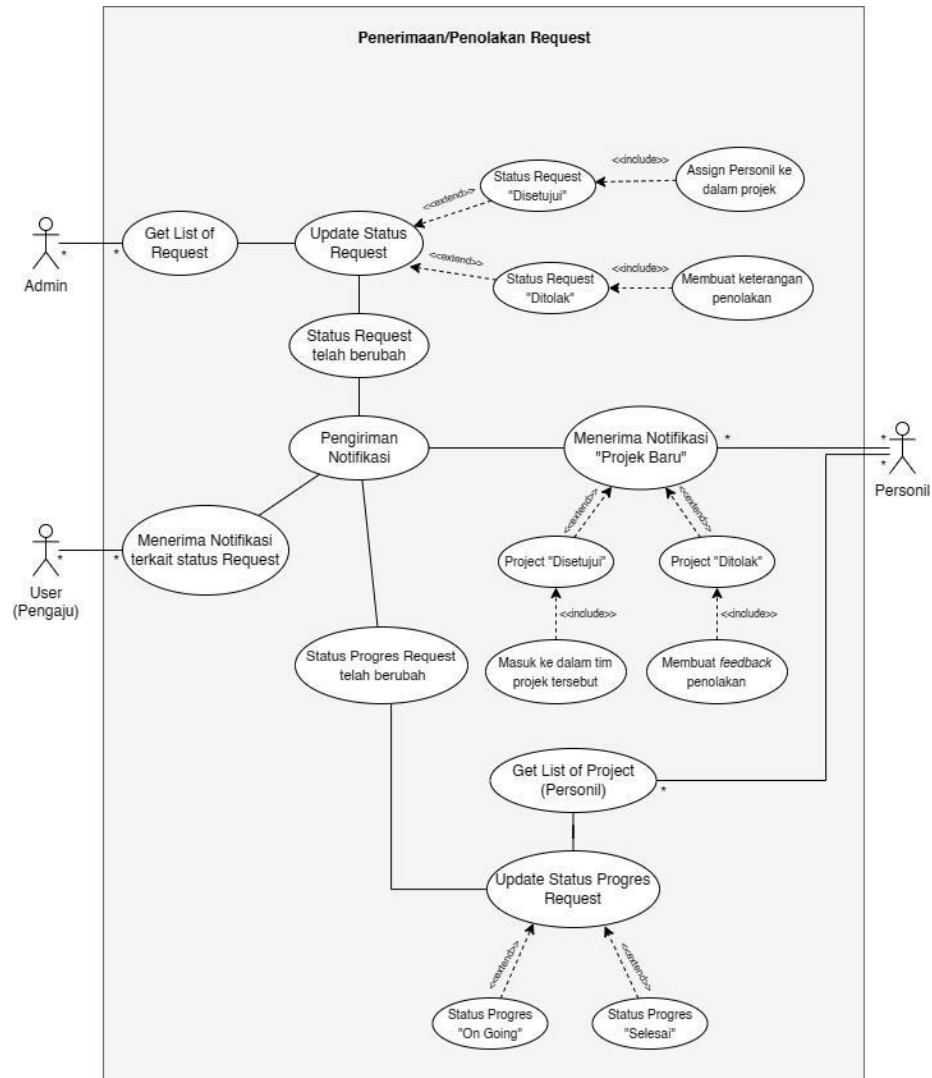
2.3 Manajemen Akun (Admin)



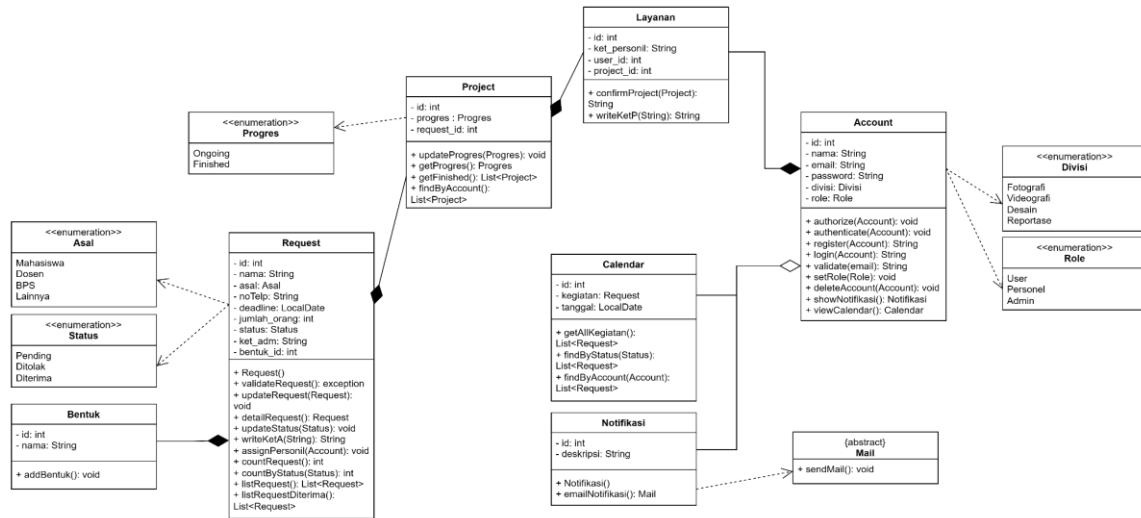
2.4 Request (Admin, User)



2.5 Penerimaan/Penolakan Request (Admin, Personel)



3. Class Diagram



Class Diagram tersebut menggambarkan atribut dan method tiap kelas. Atribut sesuai dengan data yang disimpan. Sementara itu, method sesuai dengan fungsi yang disediakan.

1. Account

Class Account mempunyai atribut id, nama, email, password, divisi, dan role. Method yang terdapat pada class ini, antara lain:

- authorize, untuk mengotorisasi akun
- authenticate, untuk mengotentikasi akun
- register, untuk mendaftarkan akun
- login, untuk masuk ke suatu akun
- validate, untuk memvalidasi email yang didaftarkan
- setRole, method yang digunakan admin untuk mengganti role suatu akun
- deleteAccount, method yang digunakan admin untuk menghapus suatu akun
- showNotifikasi, untuk menampilkan notifikasi
- viewCalendar, untuk melihat kalender kegiatan

2. Notifikasi

Class Notifikasi mempunyai atribut id dan deskripsi. Method yang terdapat pada class ini, antara lain:

- Notifikasi, untuk membuat notifikasi

- b. emailNotifikasi, untuk membuat notifikasi pada email

3. Calender

Class Calendar mempunyai atribut id, kegiatan, dan tanggal. Method yang terdapat pada class ini, antara lain:

- a. getAllKegiatan, untuk menampilkan semua kegiatan
- b. findByStatus, untuk menampilkan kegiatan berdasarkan statusnya
- c. findByAccount, untuk menampilkan kegiatan berdasarkan suatu akun

4. Layanan

Class Layanan mempunyai atribut id, ket_personil, user_id, dan project_id. Method yang terdapat pada class ini, antara lain:

- a. confirmProject, method yang digunakan personel untuk mengonfirmasi proyek
- b. writeKetP, method yang digunakan personel untuk menulis alasan penolakan proyek

5. Project

Class Project mempunyai atribut id, progres, dan request_id. Method yang terdapat pada class ini, antara lain:

- a. updateProgres, method yang digunakan personel untuk memperbarui status progres proyek
- b. getProgres, untuk mengetahui progres suatu proyek
- c. getFinished, untuk menampilkan semua proyek yang sudah selesai
- d. findByAccount, untuk mengetahui semua proyek yang ditugaskan pada seorang personel

6. Request

Class Request mempunyai atribut id, nama, asal, noTelp, deadline, jumlah_orang, status, ket_adm, dan bentuk_id. Method yang terdapat pada class ini, antara lain:

- a. Request, untuk membuat request
- b. validateRequest, untuk memvalidasi detail request yang diajukan

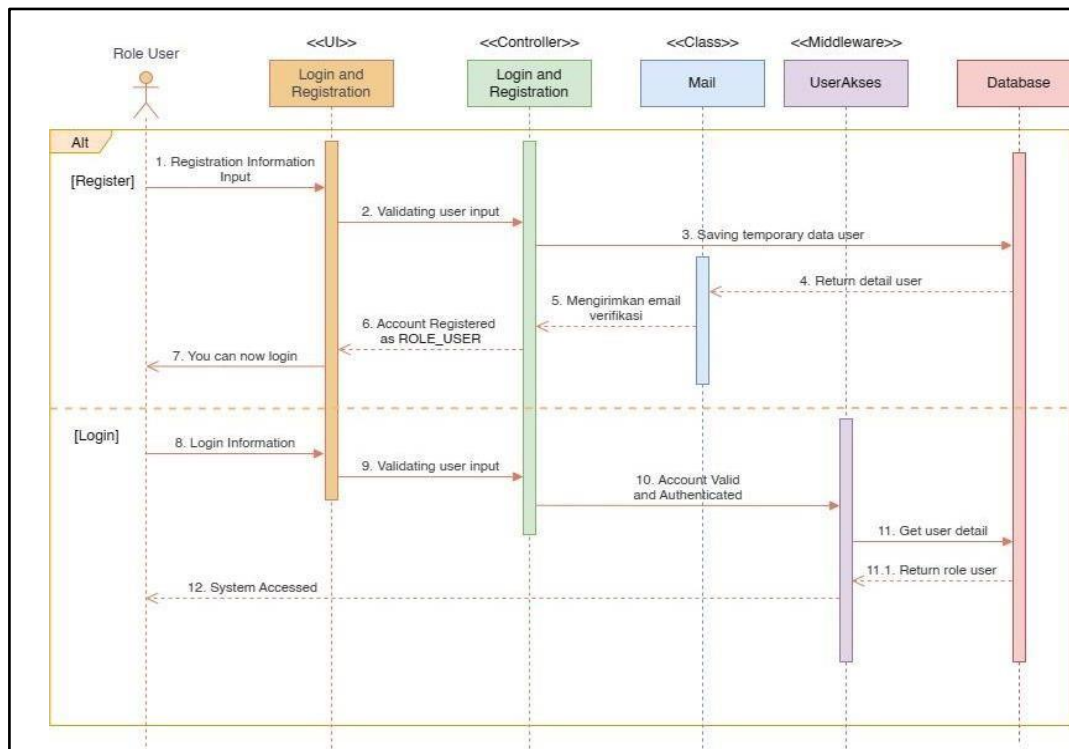
- c. `updateRequest`, untuk mengubah detail request yang diajukan
- d. `detailRequest`, untuk melihat detail request yang diajukan
- e. `updateStatus`, method yang digunakan admin untuk mengubah status suatu request
- f. `writeKetA`, method yang digunakan admin untuk menulis alasan penolakan request
- g. `assignPersonil`, method yang digunakan admin untuk menugaskan personel pada suatu request
- h. `countRequest`, untuk menghitung jumlah request
- i. `countByStatus`, untuk menghitung jumlah request berdasarkan statusnya
- j. `listRequest`, untuk menampilkan semua request
- k. `listRequestDiterima`, untuk menampilkan semua request yang telah diterima

7. Bentuk

Class Bentuk mempunyai atribut `id` dan `nama`. Method yang terdapat pada class ini adalah `addBentuk`, yang digunakan untuk menambahkan bentuk request baru.

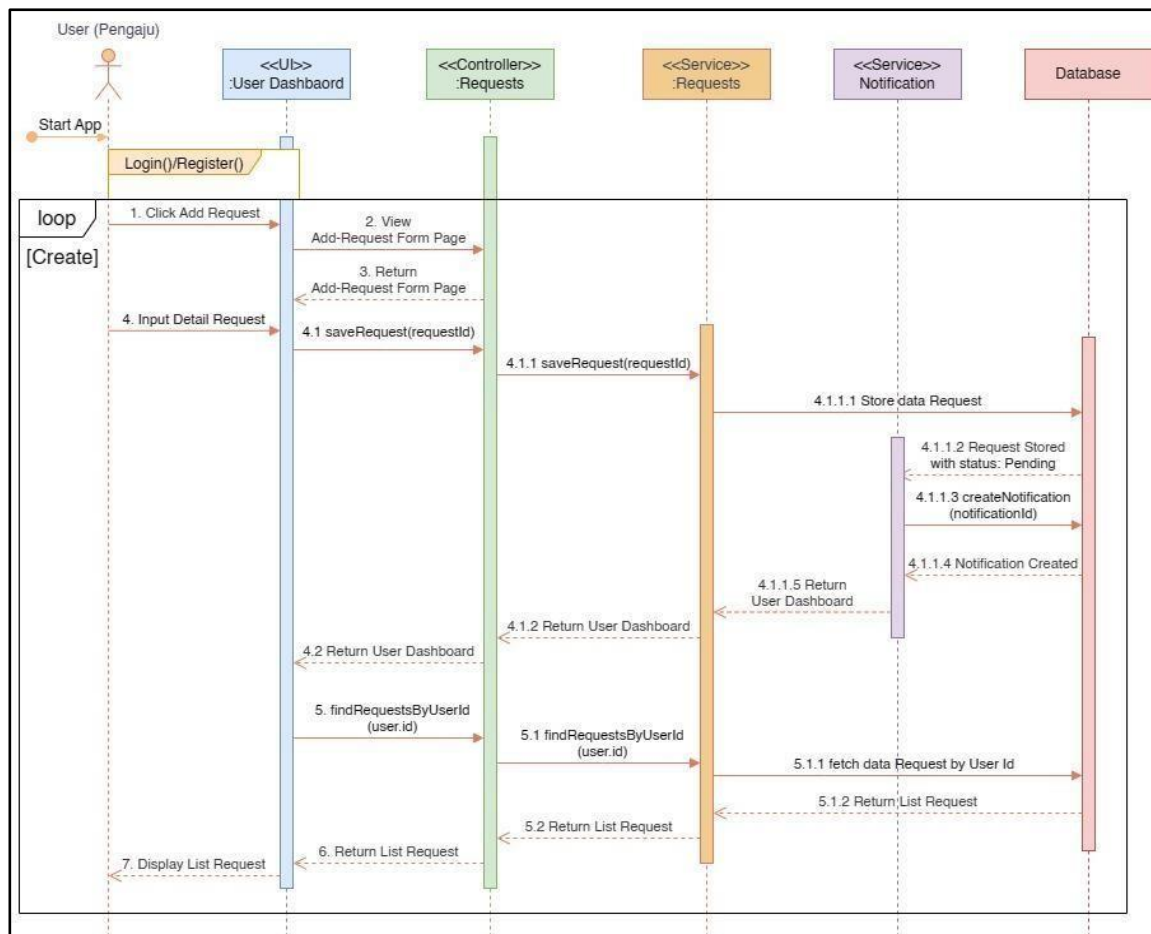
4. Sequence Diagram

1. Register dan Login (User)



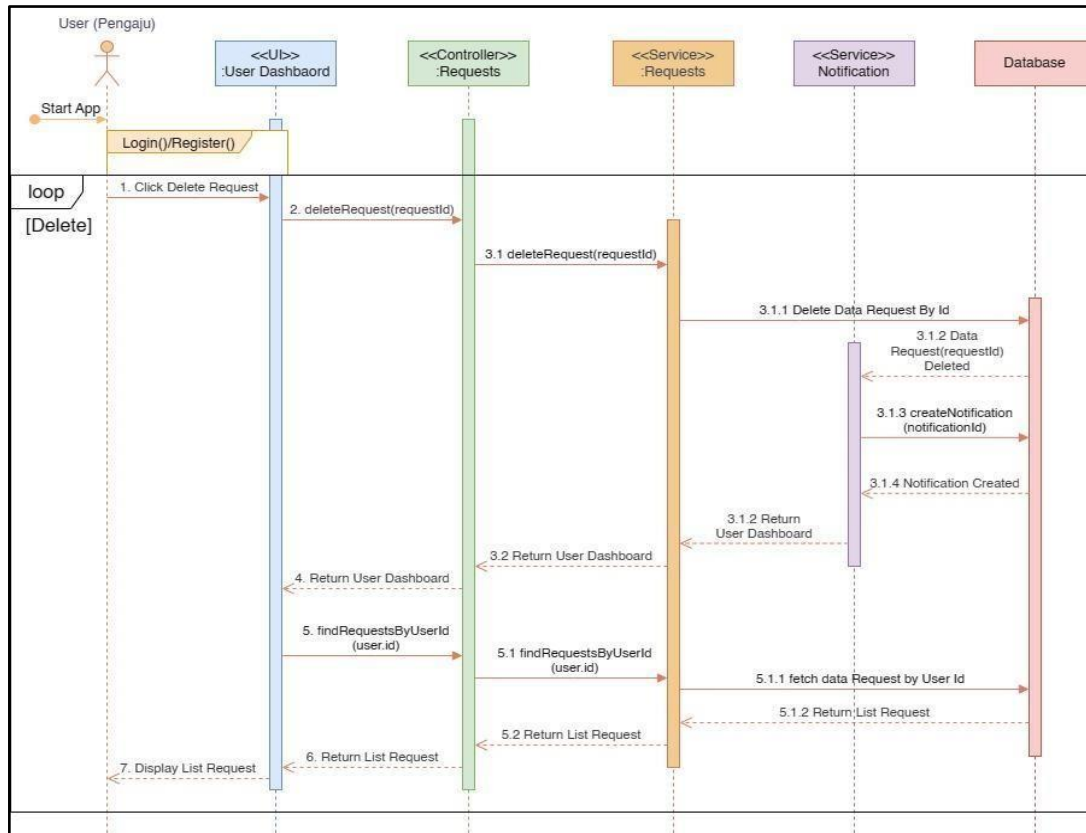
Proses bisnis yang terjadi ketika user pertama kali mengakses aplikasi web ini adalah user mengakses Login Form atau ingin mengakses Registration form. User nantinya bisa memilih untuk menggunakan “Login with Google” atau membuat akun dengan mengisi Registration form. Ketika user telah berhasil register, User akan mendapatkan email untuk verifikasi akun mereka. Jika user berhasil untuk melakukan verifikasi akun mereka, maka user dapat melakukan login dan akan diarahkan masuk ke dalam dashboard user `Sirema`.

2. Pengajuan Request (User)



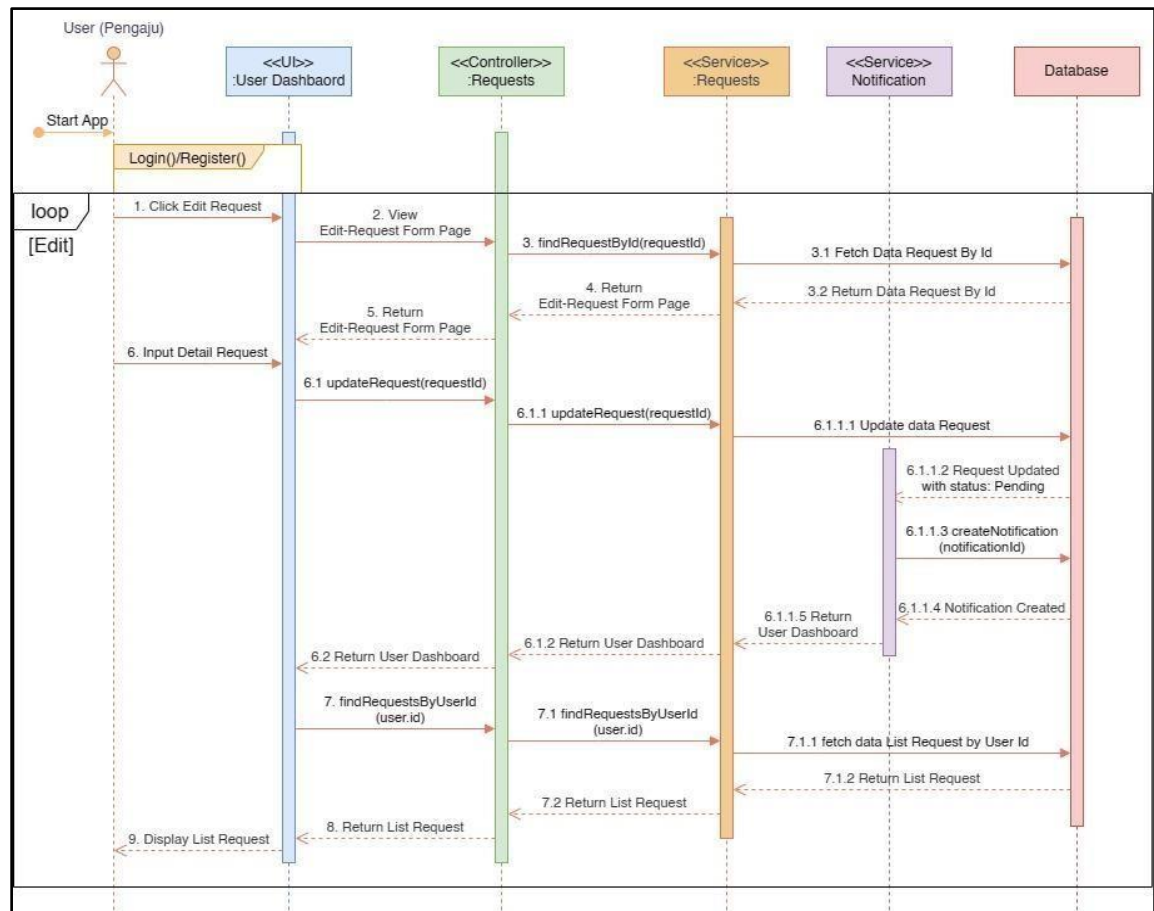
Setelah User berhasil Login, User dapat mengajukan Request baru. User dapat menekan tombol Add Request ("+"), lalu user akan diarahkan ke halaman "Add Request Form Page". Pada halaman tersebut user akan mengisi detail Request yang diajukan ke UKM Media Kampus. Setelah user berhasil membuat Request, User juga akan mendapatkan Notifikasi bahwa User telah berhasil membuat Request baru. Request baru yang telah dibuat akan tampil pada tabel dengan status "Pending".

3. Penghapusan Request (User)



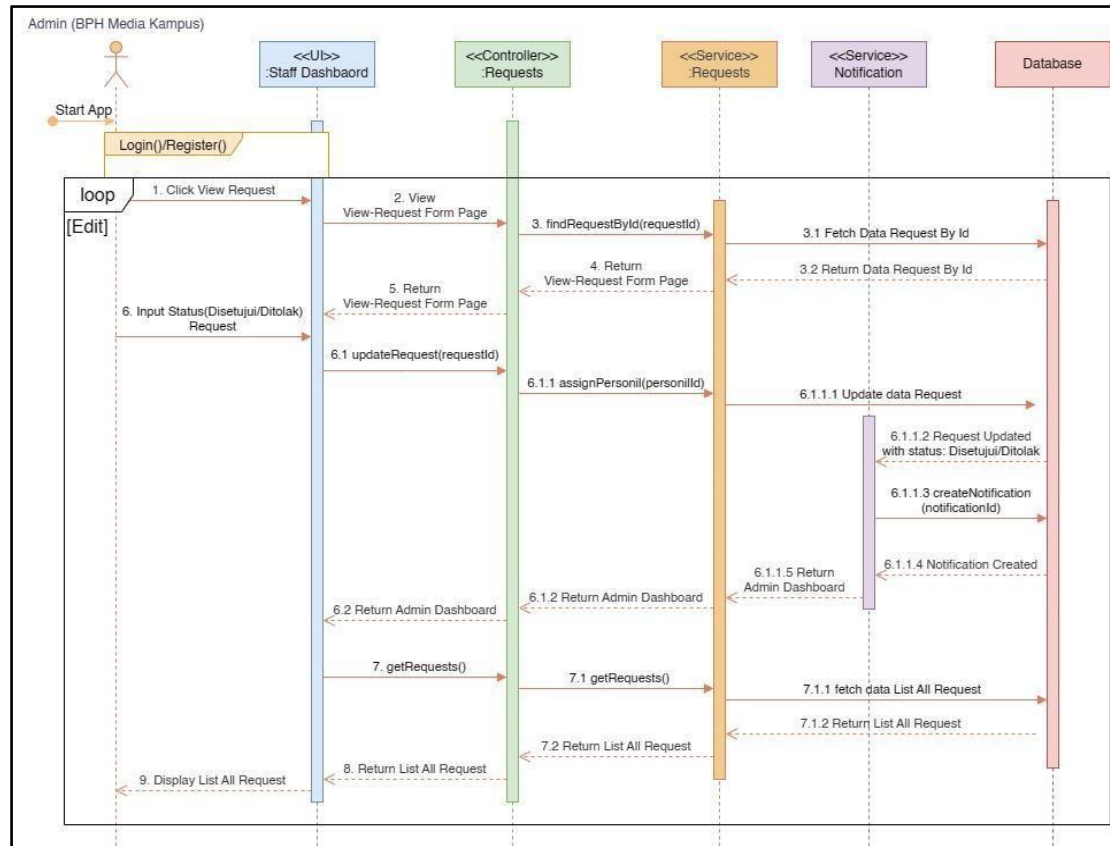
User dapat menghapus request yang telah dibuat dengan status yang masih "Pending". User dapat menekan tombol Delete Request atau icon "Trash". User akan mendapatkan notifikasi bahwa user berhasil menghapus request dari database.

4. Pengeditan Request (User)



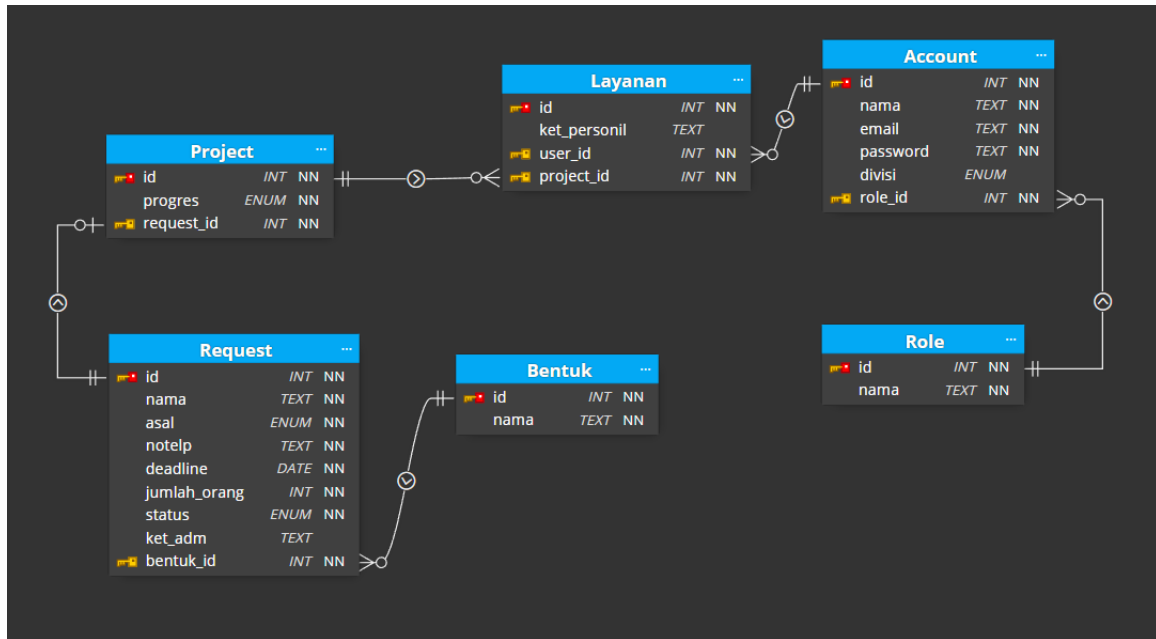
User dapat mengedit Request yang telah dibuat dengan status yang masih "Pending" atau "Ditolak". Request dengan status "Ditolak" dapat diajukan kembali dengan mengedit Requestnya. User dapat menekan tombol edit atau "Pensil" untuk mengedit Request. Setelah menekan tombol tersebut, User dapat mengisi detail Request lalu di submit kembali. Setelah berhasil mengedit Request, User akan mendapatkan Notification bahwa Request berhasil diedit/di-update.

5. Update Status Request (Admin)



Admin dapat mengubah status Request yang telah dibuat oleh User. Admin dapat menekan tombol View atau "Eye" pada Request yang tersedia. Admin tidak dapat mengubah isi dari Request, tetapi Admin hanya dapat mengubah Status Request dari "Pending" menjadi "Disetujui" atau "Ditolak". Status yang telah diubah oleh Admin, User akan mendapatkan Notification bahwa Request telah diupdate.

5. Rancangan Basis Data



Database Management System yang digunakan adalah MySQL. Terdapat 6 tabel database yang digunakan untuk memenuhi kebutuhan proses bisnis yang dilakukan, yaitu:

1. Account

Menyimpan data pengguna, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap account
- nama, menyimpan nama account
- email, menyimpan email yang didaftarkan untuk login
- password, menyimpan password yang digunakan untuk login
- divisi, menyimpan informasi divisi (fotografi, videografi, desain grafis, dan reportase & kepenulisan) account dengan role personel
- role_id, menyimpan informasi role account

2. Role

Menyimpan informasi role yang tersedia, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap role
- nama, nama role yang tersedia, yaitu user, personel, dan admin

3. Request

Menyimpan informasi mengenai request yang diajukan, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap request
- nama, menyimpan nama request yang diajukan
- asal, menyimpan informasi asal request, yaitu Mahasiswa, dosen, BPS, dan lainnya.
- notelp, menyimpan nomor telepon pengaju request
- deadline, menyimpan tanggal batas penyelesaian request
- jumlah_orang, menyimpan jumlah orang yang diperlukan untuk melakukan request
- status, menyimpan informasi request yang diajukan, yaitu pending, ditolak, dan diterima
- ket_adm, menyimpan keterangan alasan penolakan suatu request
- bentuk_id, menyimpan bentuk request, yaitu liputan, desain, penulisan naskah, pengambilan gambar dan editing video.

4. Project

Menyimpan informasi request yang telah diterima, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap project
- progres, menyimpan informasi kemajuan project, yaitu ongoing dan finished
- request_id, menyimpan informasi request mana yang disimpan

5. Layanan

Menyimpan informasi account yang terlibat dalam suatu project, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap layanan
- ket_personel, menyimpan keterangan project, yaitu diterima atau alasan penolakan suatu project
- account_id, menyimpan informasi account yang terlibat dalam project
- project_id, menyimpan informasi project mana yang dikerjakan

6. Bentuk

Menyimpan informasi bentuk request, yaitu:

- id, sebagai primary key yang mengidentifikasi tiap bentuk request
- nama, nama bentuk request yang tersedia, yaitu liputan, desain, penulisan naskah, pengambilan gambar dan editing video.

Hubungan yang terjadi antar tabel adalah:

a. Role – Account

Hubungan yang terjadi adalah *one-to-many*. Satu *account* harus dan hanya mempunyai satu *role*. Satu *role* dapat dimiliki oleh nol hingga banyak *account*.

b. Bentuk – Request

Hubungan yang terjadi adalah *one-to-many*. Satu request harus dan hanya mempunyai satu bentuk request. Satu bentuk request dapat dimiliki oleh nol hingga banyak request.

c. Request – Project

Hubungan yang terjadi adalah *one-to-one*. Satu project harus dan hanya mempunyai satu request. Satu request dapat dimiliki oleh nol hingga satu project.

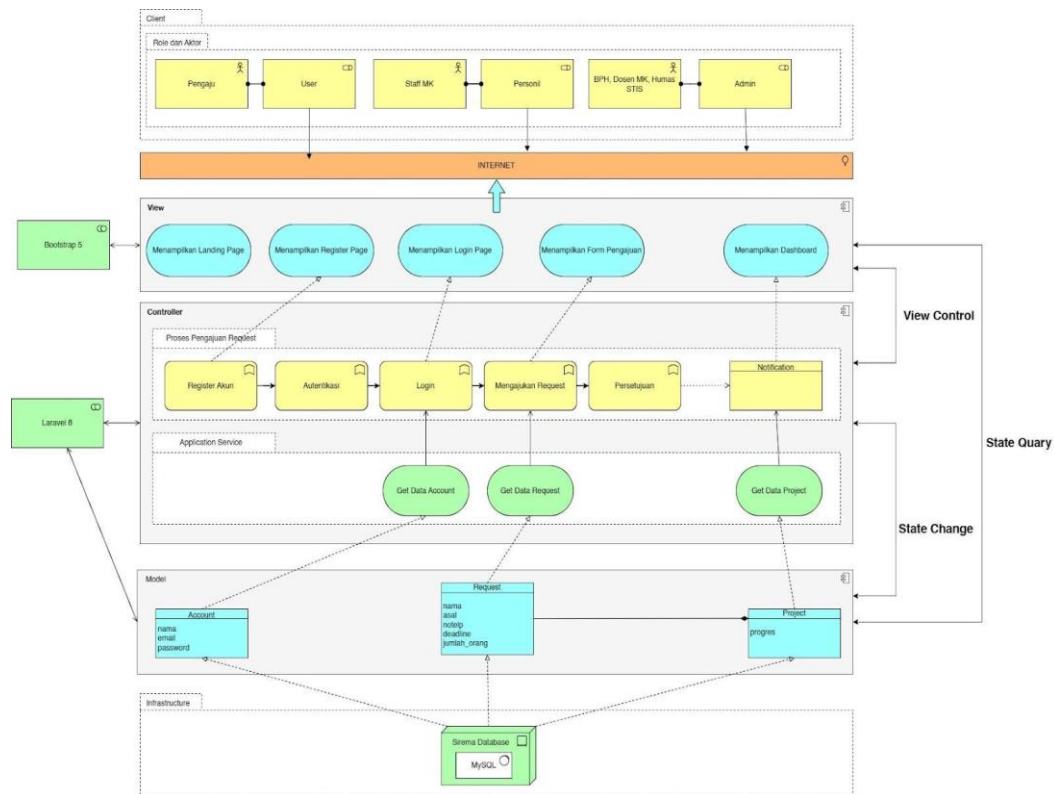
d. Project – Layanan

Hubungan yang terjadi adalah *one-to-many*. Satu layanan harus dan hanya melayani satu project. Satu project dapat dilayani oleh nol hingga banyak layanan.

e. Account – Layanan

Hubungan yang terjadi adalah *one-to-many*. Satu layanan harus dan hanya melibatkan satu account. Satu account dapat terlibat dengan nol hingga banyak layanan.

6. Rancangan Arsitektur Sistem



Rancangan arsitektur sistem yang tim kami bangun menggunakan model MVC (Model-View-Controller). Penggunaan model arsitektur MVC dikarenakan penggunaan arsitektur ini akan memudahkan dalam hal pengembangan, pemeliharaan, serta pengujian aplikasi website yang tim kami sedang bangun. Model arsitektur MVC juga memberikan kemudahan bagi tim backend dan frontend kami dalam bekerja sama menjadi lebih fleksibel dan dapat dilakukan secara terpisah. Penggunaan MVC juga menyesuaikan dengan framework yang tim kami gunakan yaitu Laravel serta server Politeknik Statistika STIS yang mendukung model arsitektur ini. Pada rancangan arsitektur sistem kami akan terbagi menjadi 3 komponen utama yaitu sisi client, MVC, dan infrastruktur.

Sisi client terdapat 3 aktor utama yang akan menggunakan serta menjalankan aplikasi website Sirema ini yaitu User, personel, dan Admin. Aktor pertama adalah Pengaju dimana Pengaju tersebut akan mendapatkan role “User”. Pengaju ini adalah client UKM Media Kampus yang akan melakukan pengajuan suatu *Request* atau *Project* yang membutuhkan tenaga dari Media Kampus. Aktor kedua adalah para staff Media Kampus dimana staff Media

Kampus tersebut akan mendapatkan role “personel”. personel ini akan melakukan perubahan status progres dari *Project* atau *Request* yang sudah disetujui oleh Admin. Aktor ketiga adalah BPH (Badan Pengurus Harian) Media Kampus, Dosen Pembina Media Kampus, dan HUMAS STIS akan mendapatkan role “Admin”. Admin ini akan mengelola semua request yang diajukan ke Media Kampus, Mengelola personel agar pembagian tugas di dalam Media Kampus dapat terbagi menjadi rata, serta dapat mengelola akun-akun yang terdaftar di website Sirema. Client ini dapat mengakses website sirema dari device apapun dengan syarat terhubung dengan internet.

Komponen selanjutnya yaitu komponen MVC. Komponen MVC ini terbagi menjadi 3 yaitu komponen Model, View, Controller. Komponen view ini bertanggung jawab dengan bagaimana data dapat ditampilkan kepada pengguna. Komponen ini akan mengambil data dari model dan memiliki beberapa logika tampilan untuk menentukan bagaimana data harus ditampilkan seperti menampilkan dashboard akan berbeda untuk ketiga role serta data yang akan ditampilkan berbeda juga. View merupakan bagian yang dapat diintegrasikan oleh pengguna nantinya seperti mengisi form, melakukan filter terhadap data, dan sebagainya. Komponen Controller bertindak sebagai perantara antara View dan Model. Controller akan menerima input dari pengguna melalui komponen View kemudian mengambil tindakan yang diperlukan berdasarkan input tersebut. Hubungan antara View dan Controller tersebut disebut **View Control**. Dalam memproses suatu input atau aksi dari pengguna, Controller akan memberi tahu model terkait perubahan yang terjadi. Hubungan antara Controller dengan Model disebut dengan **State Change**. Komponen Model dalam arsitektur MVC merepresentasikan data yang digunakan pada aplikasi. Model akan bertindak sebagai sumber data dan status yang dapat di-*query* oleh View. Model tidak memiliki referensi langsung ke View, namun dapat mengirimnya ke Controller saat data berubah yang kemudian menginstruksikan View untuk memperbarui tampilannya. Hubungan antara Model dengan View ini disebut dengan **State Query**. Sebagai contoh, ketika seorang pengguna berhasil melakukan pengajuan suatu request maka Model akan mengambil data dari Model dan memprosesnya lalu akan memberitahu View untuk menampilkan data yang telah diproses. Terdapat tiga entitas utama dalam komponen Model yaitu Account, Request, dan Project. Dalam menyimpan suatu data di dalam Model diperlukan infrastruktur yang tepat.

Infrastruktur yang digunakan pada rancangan arsitektur website Sirema terbagi menjadi 3 yaitu infrastruktur untuk *Frontend*, *Backend*, serta *Database*. Infrastruktur yang digunakan untuk *Backend* adalah Laravel versi 8. Penggunaan Laravel versi 8 dikarenakan server pada Politeknik Statistika STIS yang mendukung Laravel versi 8. Penggunaan framework Laravel ini juga membuat pemilihan infrastruktur untuk *Database* menggunakan MySQL. Penggunaan MySQL ini juga dikarenakan server yang akan digunakan mendukung penggunaan tersebut. Infrastruktur untuk *Frontend* adalah Bootstrap 5. Penggunaan Bootstrap 5 dikarenakan anggota dari tim pengembang website Sirema sudah terbiasa dengan penggunaan infrastruktur tersebut serta dukungan terhadap infrastruktur *Backend* yang digunakan.

7. Design Pattern

Design pattern yang digunakan dalam pengembangan sistem ini diantaranya:

a. Singleton Pattern

Singleton Pattern adalah desain pola yang memastikan bahwa sebuah kelas memiliki hanya satu instance dan menyediakan titik akses global ke instance tersebut. Dalam Laravel, pola Singleton memastikan bahwa hanya ada satu instance kelas, membantu proyek tetap terorganisir dengan memastikan bahwa data konsisten di seluruh aplikasi dan pembagian resource teratur. Penerapan singleton pattern dalam pengembangan sistem ini diantaranya dalam pengelolaan request mempunyai requestController yang hanya mengelola modul permintaan jasa, userController yang hanya mengelola dashboard user, authController yang hanya mengelola autentikasi dan otorisasi user.

b. Observer Pattern

Observer Pattern adalah salah satu desain pola (design pattern) dalam pemrograman berorientasi objek yang digunakan untuk mengimplementasikan mekanisme pemberitahuan antara objek-objek yang terlibat. Desain pola ini memastikan bahwa ketika satu objek mengalami perubahan keadaan, akan dikirim notifikasi ke objek yang lain. Penerapan observer pattern dalam pengembangan sistem ini diantaranya real time notifikasi dalam web yaitu user akan menerima notifikasi saat admin menyetujui request, menerima email verifikasi saat user registrasi web.

c. Facade Pattern

Facade Pattern adalah pola desain struktural yang menyediakan antarmuka tingkat tinggi yang memudahkan penggunaan atau integrasi dengan sistem yang lebih kompleks atau kumpulan antarmuka yang lebih banyak. Ini menyembunyikan kompleksitas sistem yang berada di belakangnya dan menyediakan titik akses tunggal untuk penggunaan atau integrasi. Penerapan facade pattern dalam pengembangan sistem ini yaitu tampilan interface setiap role berbeda.

d. Repository Pattern

Repository Pattern adalah suatu pendekatan yang memisahkan antara Business Logic layer dengan Data Access Logic layer. Menggunakan repository pattern, business logic layer tidak harus tahu dari mana sumber data diambil atau ke mana data akan dikirim. Business logic layer hanya bertugas untuk melakukan implementasi proses bisnis yang ingin diselesaikan atau masalah yang seharusnya diselesaikan. Penerapan repository pattern dalam pengembangan sistem ini diantaranya penggunaan class model dalam membentuk objek di dalam database.