**Requirement analysis (FURPS)**

Entitas, actors :

(mahasiswa, dosen, uaa ,operator)

**Functionality :**

- **Capability :** sistem harus mendukung peran pengguna yang berbeda (mahasiswa, dosen, uaa ,operator) dengan fungsi peran tertentu. Bagi mahasiswa, ini mungkin termasuk melihat nilai, melihat data pribadi, Jadwal matakuliah, Registrasi krs/Memilih krs, dan Melihat ipk/ips. Dosen Memberi nilai, Mengabsen mahasiswa, Melihat anggota kelas dan Mengupload soal ujian. Uaa Mengupload jadwal, Mengupload krs/ mengupload registrasi krs, Memberitahu detail pertemuan kelas, Mengupload ipk mahasiswa, admin support dosen  dukungan teknis

- **Reusability :** komponen sistem sisfo dirancang dengan cara yang memungkinkan penggunaan kembali di berbagai bagian sistem atau bahkan di proyek perangkat lunak lainnya

- **Security :** Sistem memiliki mekanisme keamanan yang memadai untuk melindungi data sensitif seperti informasi informasi pribadi, akademik. Pengguna harus diotentikasi dengan benar sebelum diizinkan mengakses fitur-fitur tertentu, seperti registrasi krs atau mengakses informasi akun.

**Usability :**

- **Human Factors :** Antarmuka pengguna sistem sisfo dirancang dengan memperhatikan kebutuhan dan keterbatasan pengguna, Proses pendaftaran krs, detail nilai, dan jadwal mudah dipahami.

- **Consistency :** konsisten dalam hal desain antarmuka pengguna, terminologi, dan alur kerja di seluruh platform. Ini akan memungkinkan pengguna untuk dengan mudah beradaptasi dengan sistem dan meminimalkan kebingungan. diperkenalkan dalam sistem, seperti pembaruan fitur atau penyesuaian desain.

- **Documentation :** dokumentasi mencakup informasi tentang bagaimana mengakses dan menggunakan fitur-fitur sistem, langkah-langkah untuk melakukan tugas-tugas tertentu seperti pendaftaran krs atau pengelolaan profil mahasiswa.

**Reliability :**

- **Availability :** istem selalu tersedia 24/7, dengan waktu henti minimal, pengguna dapat mengakses layanan sistem kapan pun mereka membutuhkan
- **Failure Rate & Duration :** tingkat kegagalan yang rendah, dengan sedikit atau tidak ada gangguan yang menyebabkan sistem tidak berfungsi.
- **Predictability :** Perilaku sistem, seperti waktu respon dan ketersediaan layanan, akan konsisten dan dapat diandalkan dari waktu ke waktu

**Performance :**

- **Speed :** Kecepatan respons sistem terhadap permintaan pengguna menampilkan informasi atau menanggapi tindakan, seperti menampilkan jadwal perkuliahan, mengambil data mahasiswa, atau memproses permintaan registrasi.
- **Efficiency** : Sistem dirancang untuk menjalankan tugas-tugasnya dengan menggunakan sejumlah minimum sumber daya yang memungkinkan.
- **Resource Consumption :** Sistem memantau dan mengelola konsumsi sumber daya dengan cermat, konsumsi sumber daya harus dilakukan secara terus-menerus untuk mendeteksi dan menangani masalah yang timbul segera.
- **Scalability :** Sistem sisfo dapat menangani peningkatan beban pengguna dengan baik seiring waktu, tanpa mengalami penurunan kinerja atau gangguan layanan.

**Supportability :**

- **Testability :** Sistem perpustakaan dirancang agar mudah diuji untuk memastikan kualitas dan keandalan kode serta fungsionalitas sistem secara keseluruhan.
- **Extensibility :** Sistem dirancang agar mudah diperluas dengan menambahkan fitur baru atau mengintegrasikan modul tambahan.
- **Serviceability :** sistem akan mudah untuk dipelihara dan diperbaiki ketika terjadi masalah atau ditemukan kekurangan.
- **Configurability :** Sistem sisfo dapat dikonfigurasi sesuai dengan kebutuhan dan preferensi pengguna.

**Use case story**

User roles :

1. mahasiswa
2. Dosen
3. Uaa
4. operator

User story

- Mahasiswa
    - Bisa melihat data pribadi
    - Melihat Nilai mata kuliah
    - Registrasi krs/Memilih krs
    - Jadwal matakuliah
    - Melihat ipk/ips
- Dosen
    - Memberi nilai
    - Mengabsen mahasiswa
    - Melihat anggota kelas
    - Mengupload soal ujian

- Uaa
    - Mengupload jadwal
    - Mengupload krs/ mengupload registrasi krs
    - Mengupload ipk mahasiswa
- operator
    - support dosen
    - dukungan teknis

**class diagram :**

- main classes: user,profile, registrasi,learning,
- user class : mahasiswa,dosen,uaa,operator
- relationships :
    - user to profile(melihat tagihan dan ipk)
    - user to registrasi (memilih dan mendaftar)
    - user to learning(melihat,mendownload )
- atribut : khusus untuk setiap kelas, matakuliah,ruangan,catatan nilai, rincian tagihan
- metode :
- user
- attributes : useriD,name,email,password
- methods : login(), logout()

mahasiswa

- attributes : pendaftaranID,nilai(daftar),absen,matakuliah(daftar),krs(daftar)
- methods : lihatDataPribadi() lihatNilaiMataKuliah() registrasiKRS() lihatJadwalMataKuliah() lihatIPK()

dosen

- attributes : dosenD, matakulaih yang diajarkan (daftar)
- methods : memberiNilai(), mengabsenMahasiswa(), lihatAnggotaKelas(), uploadSoalUjian()

UAA

- attributes : UAAID
- methods : uploadJadwal(), uploadKRS(), uploadIPKMahasiswa()

Operator

- attributes : OperatorID
- methods : supportDosen() , dukunganTeknis()

Profile

- attributes : userid, tagihan, ipk
- methods : lihatTagihan(), lihatIPK()

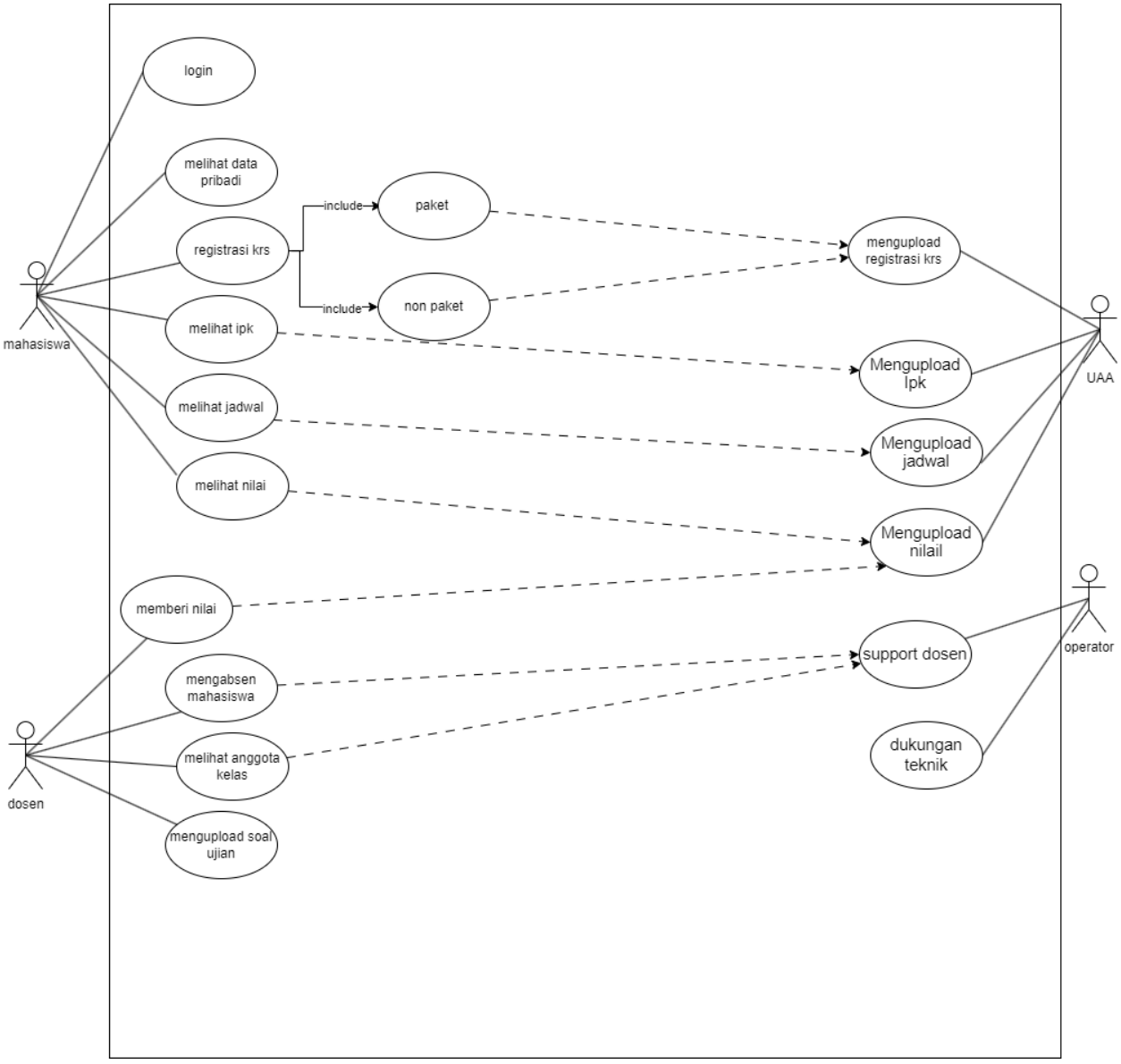
Registrasi

- attributes : registrasiID, matakuliah, ruangan
- methods : memilihmatakuliah(), mendaftarMataKuliah()

Learning

- attributes : UserID, Materi,
- methods : melihatMateri(), mendownloadMateri(),

**use case diagram :**



**Class diagram :**