

**UJIAN TENGAH SEMESTER (UTS)
GANJIL 2025/2026**

Mata Kuliah:
PEMROGRAMAN BERORIENTASI OBJEK

Kelas D3TI 2A

Studi Kasus: Aplikasi StudyTrack Berbasis Console



Oleh:
Alfath Shafira
2403026

**D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
OKTOBER 2025**

Daftar Isi

| | |
|---|-----------|
| Deskripsi Permasalahan | 1 |
| Analisis Kebutuhan | 1 |
| Analisis Fitur | 1 |
| Rancangan UML Use Case Diagram | 3 |
| Rancangan Tampilan Program Aplikasi Console..... | 4 |
| Tampilan Login Mahasiswa (Berhasil) | 4 |
| Tampilan Login Mahasiswa (Gagal) | 4 |
| Tampilan Login Dosen (Berhasil) | 4 |
| Tampilan Login Dosen (Gagal) | 4 |
| Tampilan Menu Utama (Mahasiswa) | 4 |
| Tampilan Menu Utama (Dosen) | 4 |
| Tampilan Menu Mahasiswa (Jadwalkan Sesi Berhasil) | 5 |
| Tampilan Menu Dosen (Tambah Jadwal Berhasil) | 5 |
| Tampilan Menu Mahasiswa (Lihat Jadwal Berhasil)..... | 5 |
| Tampilan Menu Dosen (Tambah Topik Berhasil)..... | 5 |
| Tampilan Menu Mahasiswa (Tandai Selesai Berhasil)..... | 5 |
| Tampilan Menu Dosen (Buat Kuis Berhasil) | 5 |
| Tampilan Menu Mahasiswa (Ambil Kuis Berhasil) | 6 |
| Tampilan Menu Dosen (Lihat Semua Jadwal Berhasil) | 6 |
| Tampilan Menu Mahasiswa (Lihat Progress Berhasil)..... | 6 |
| Tampilan Menu Mahasiswa (Lihat Riwayat Berhasil)..... | 6 |
| Tampilan Menu Mahasiswa (Jadwalkan Sesi Belum Berhasil) | 6 |
| Tampilan Menu Dosen (Tambah Topik Gagal)..... | 6 |
| Tampilan Menu Mahasiswa (Tandai Selesai Belum Berhasil) | 7 |
| Tampilan Menu Mahasiswa (Ambil Kuis Belum Berhasil) | 7 |
| Tampilan Menu Mahasiswa (Lihat Progress Belum Berhasil) | 7 |
| Tampilan Menu Mahasiswa (Lihat Riwayat Belum Berhasil) | 7 |
| Analisis Class..... | 7 |
| Identifikasi Class dan Attribute | 7 |
| Identifikasi Method | 8 |
| Jenis Relasi dan Alasan..... | 9 |
| Multiplicity | 11 |
| Rancangan UML Class Diagram | 12 |
| Rancangan UML Sequence Diagram | 13 |

| | |
|---|-----------|
| Sequence Diagram: Login..... | 13 |
| Sequence Diagram: Ambil Kuis | 14 |
| Sequence Diagram: Buat Kuis | 15 |
| Sequence Diagram: Jadwal Sesi & Tandai Selesai | 16 |
| Sequence Diagram: Jadwal Sesi & Tandai Selesai | 17 |
| Sequence Diagram: Progress & Riwayat | 18 |
| Kode Program | 19 |
| Aktivitas.java | 19 |
| Dosen.java..... | 19 |
| HasilKuis.java..... | 20 |
| Jadwal.java | 21 |
| Kuis.java | 22 |
| Mahasiswa.java | 23 |
| Main.java..... | 24 |
| Pengguna.java | 26 |
| SesiBelajar.java | 27 |
| Soal.java | 28 |
| StudyTrackApp.java | 29 |
| Topik.java | 34 |
| Compile & Run..... | 35 |
| Compile | 35 |
| Run | 35 |
| Testing..... | 35 |
| Skenario Login Mahasiswa Valid (Berhasil) | 35 |
| Skenario Login Mahasiswa Tidak Valid (Berhasil) | 35 |
| Skenario Login Dosen Valid (Berhasil)..... | 36 |
| Skenario Login Dosen Tidak Valid (Berhasil) | 36 |
| Skenario Pilih Menu Mahasiswa Tidak Valid (Berhasil)..... | 36 |
| Skenario Pilih Menu Dosen Tidak Valid (Berhasil)..... | 36 |
| Skenario Pilih Menu Jadwalkan Sesi Tidak Ditemukan (Berhasil) | 36 |
| Skenario Pilih Menu Jadwalkan Sesi (Berhasil) | 37 |
| Skenario Pilih Menu Lihat Jadwal (Berhasil) | 37 |
| Skenario Pilih Menu Tandai Selesai Tidak Ditemukan (Berhasil) | 37 |
| Skenario Pilih Menu Tandai Selesai (Berhasil)..... | 37 |
| Skenario Pilih Menu Ambil Kuis Tidak Ditemukan (Berhasil) | 37 |
| Skenario Pilih Menu Ambil Kuis (Berhasil) | 38 |

| | |
|--|-----------|
| Skenario Pilih Menu Lihat Progress (Berhasil) | 38 |
| Skenario Pilih Menu Lihat Riwayat (Berhasil)..... | 38 |
| Skenario Pilih Menu Lihat Progress Tanpa Data (Berhasil)..... | 39 |
| Skenario Pilih Menu Lihat Riwayat Tanpa Data (Berhasil)..... | 39 |
| Skenario Pilih Menu Dosen Tambah Topik Tidak Ditemukan (Berhasil)..... | 39 |
| Skenario Pilih Menu Dosen Tambah Jadwal (Berhasil) | 39 |
| Skenario Pilih Menu Dosen Tambah Topik (Berhasil)..... | 39 |
| Skenario Pilih Menu Dosen Buat Kuis (Berhasil) | 40 |
| Skenario Pilih Menu Dosen Lihat Semua Jadwal (Berhasil) | 40 |
| Build (Deploy) | 40 |
| URL Repository | 40 |

Deskripsi Permasalahan

Aplikasi **StudyTrack** adalah sistem berbasis console yang dirancang untuk membantu mahasiswa dalam mengelola aktivitas belajar secara mandiri dan terarah. Program ini memudahkan mahasiswa untuk menjadwalkan sesi belajar, menandai sesi yang telah diselesaikan, mengikuti kuis, serta memantau progres dan riwayat pembelajaran secara langsung. Di sisi lain, dosen dapat menambahkan kursus, topik, dan kuis yang akan dikerjakan oleh mahasiswa. Dengan sistem ini, proses pencatatan waktu belajar, nilai kuis, serta aktivitas belajar menjadi lebih terstruktur dan mudah dipantau melalui antarmuka console yang sederhana.

Analisis Kebutuhan

Adanya aplikasi **StudyTrackApp** berbasis console yang digunakan untuk mengelola aktivitas pembelajaran mahasiswa. Mahasiswa dapat login, menjadwalkan sesi belajar, menandai sesi yang telah selesai, mengikuti kuis, serta melihat progres dan riwayat aktivitasnya. Dosen dapat login untuk menambahkan kursus, topik, dan kuis yang akan dikerjakan oleh mahasiswa. Seluruh data aktivitas tersimpan dan dikelola oleh sistem melalui antarmuka console yang sederhana. Analisis kebutuhan program ini terdiri dari:

1. Dosen dapat menambah kursus, menambah topik, serta membuat kuis yang berisi daftar soal pilihan ganda.
2. Mahasiswa dapat menjadwalkan sesi belajar berdasarkan kursus dan topik yang tersedia.
3. Mahasiswa dapat melihat daftar jadwal sesi belajar yang telah dibuat.
4. Mahasiswa dapat menandai sesi belajar sebagai selesai, dan sistem akan otomatis menambah total menit belajar ke profil student.
5. Mahasiswa dapat mengambil kuis, menjawab soal, dan mendapatkan nilai akhir.
6. Mahasiswa dapat melihat progres belajar, meliputi total waktu belajar, jumlah topik yang telah diselesaikan, serta rata-rata nilai kuis.
7. Mahasiswa dapat melihat riwayat aktivitas, baik sesi belajar maupun hasil kuis yang telah diselesaikan.

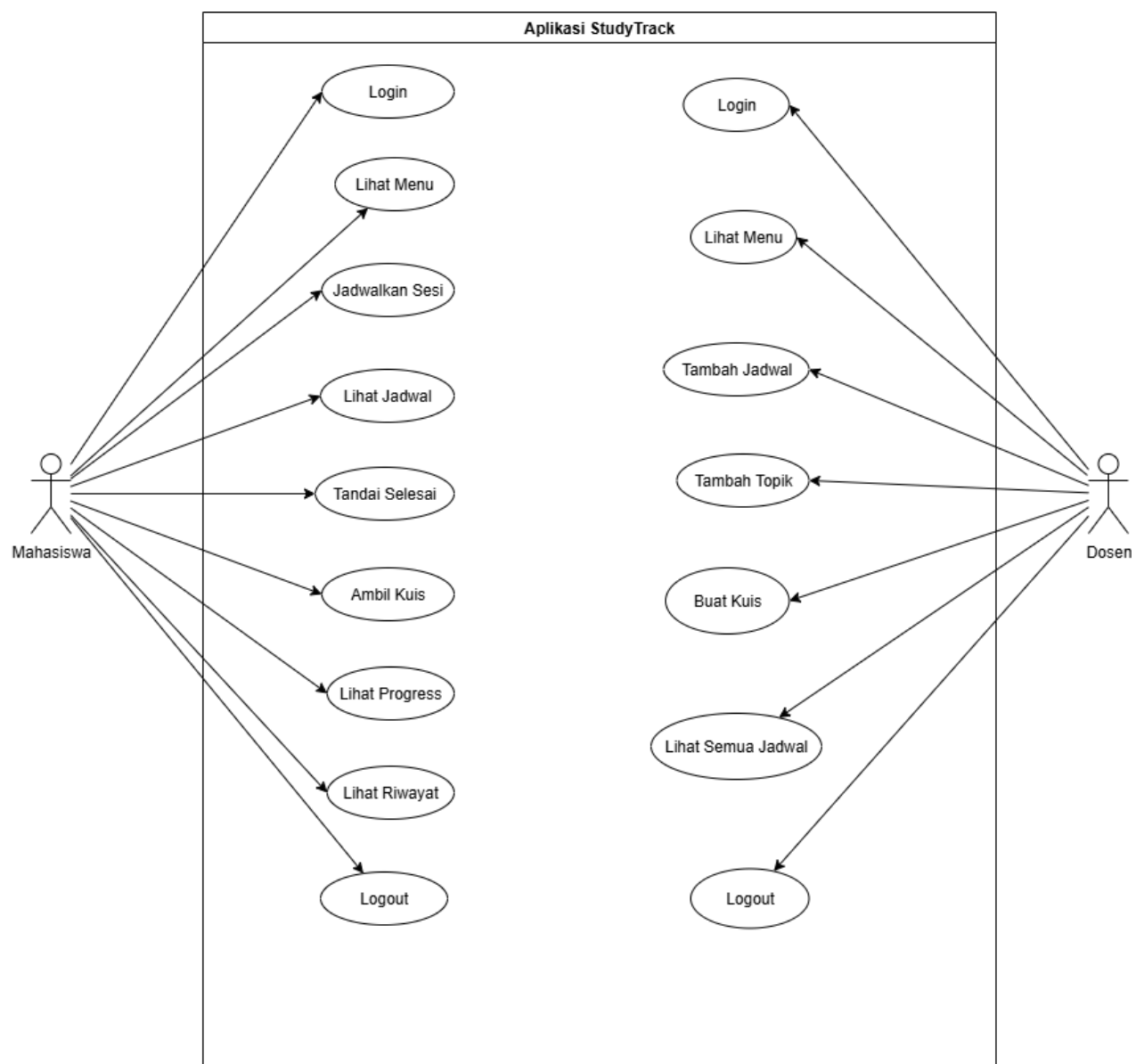
Analisis Fitur

| No | Fitur | Kebutuhan | Alur |
|----|------------------------------------|---|---|
| 1 | Login Pengguna (Mahasiswa & Dosen) | Aplikasi perlu membedakan hak akses berdasarkan peran pengguna. | Pengguna memilih peran (Mahasiswa atau Dosen), lalu memasukkan username dan password untuk diverifikasi. |
| 2 | Menampilkan Menu Utama (Mahasiswa) | Setelah login berhasil, ditampilkan menu dengan opsi: 1. Jadwalkan Sesi 2. Lihat Jadwal 3. Tandai Selesai 4. Ambil Kuis 5. Lihat Progress 6. Lihat Riwayat 7. Logout | Mahasiswa memilih opsi, dan program mengarahkan ke fungsi terkait. Menu berulang hingga mahasiswa memilih keluar. |

| No | Fitur | Kebutuhan | Alur |
|----|--------------------------------------|--|---|
| 3 | Menampilkan Menu Utama (Dosen) | Setelah login berhasil, ditampilkan menu dengan opsi: 1. Tambah Kursus 2. Tambah Topik 3. Buat Kuis 4. Lihat Semua Kursus 5. Logout | Dosen memilih opsi, dan program mengarahkan ke fungsi terkait. Menu berulang hingga dosen memilih keluar. |
| 3 | Manajemen Kursus (Dosen) | Dosen perlu menambah kursus dan mendefinisikan deskripsi materi. | Dosen login → pilih menu “tambah kursus” → input ID kursus, nama kursus, dan deskripsi kursus. |
| 3 | Manajemen Topik (Dosen) | Setiap kursus dapat memiliki beberapa topik pembelajaran. | Dosen login → pilih menu “tambah topik” → input topik dengan ID kursus tujuan, ID topik, judul topik, dan keterangan. |
| 4 | Pembuatan Kuis (Dosen) | Dosen perlu membuat kuis berisi soal untuk setiap topik. | Dosen login → pilih menu “buat kuis” → input ID topik, ID kuis, berapa pertanyaan, pilihan jawaban, dan kunci jawaban. |
| 5 | Melihat Daftar Semua Kursus (Dosen) | Dosen ingin mengetahui daftar semua kursus dan topik yang tersedia. | Dosen login → pilih menu “Lihat Semua Kursus” → sistem menampilkan daftar semua kursus beserta topik di dalamnya. |
| 6 | Penjadwalan Sesi Belajar (Mahasiswa) | Mahasiswa perlu membuat jadwal belajar untuk topik tertentu. | Mahasiswa login → pilih menu “jadwalkan sesi” → input ID kursus, ID topik, tanggal dan durasi → sesi disimpan ke dalam daftar. |
| 7 | Melihat Jadwal (Mahasiswa) | Mahasiswa ingin mengetahui jadwal dan topik yang diambil. | Mahasiswa login → pilih menu “Lihat Jadwal” → sistem menampilkan daftar jadwal beserta topik yang diambil. |
| 8 | Menandai Sesi Selesai (Mahasiswa) | Mahasiswa ingin mencatat sesi belajar yang telah diselesaikan. | Mahasiswa login → pilih menu “tandai selesai” → input ID jadwal kemudian sistem mencatat aktivitas yang telah diselesaikan. |
| 9 | Mengambil Kuis (Mahasiswa) | Mahasiswa ingin mengerjakan kuis dan memperoleh nilai. | Mahasiswa login → pilih menu “ambil kuis” → input ID kuis → jawab pertanyaan → sistem menghitung nilai → hasil disimpan ke riwayat. |
| 10 | Melihat Progress Belajar (Mahasiswa) | Mahasiswa ingin mengetahui sejauh mana perkembangan belajarnya. | Mahasiswa login → pilih menu “lihat progress” → sistem menampilkan total waktu belajar, topik selesai, dan rata-rata nilai kuis. |

| No | Fitur | Kebutuhan | Alur |
|----|---------------------------------------|---|---|
| 11 | Melihat Riwayat Aktivitas (Mahasiswa) | mahasiswa ingin meninjau aktivitas belajar dan kuis yang telah dilakukan. | Mahasiswa login → pilih menu “lihat riwayat” → sistem menampilkan daftar aktivitas dengan tanggal dan jenisnya. |

Rancangan UML Use Case Diagram



Rancangan Tampilan Program Aplikasi Console

Tampilan Login Mahasiswa (Berhasil)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen  
(3)Logout: 1  
Username: alshfra  
Password: mhs1910  
Login berhasil. Halo, Alfath Shafira (Mahasiswa)
```

Tampilan Login Mahasiswa (Gagal)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen  
(3)Logout: 1  
Username: mahasiswa  
Password: mhs  
Login gagal! Silahkan coba lagi.
```

Tampilan Login Dosen (Berhasil)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen  
(3)Logout: 2  
Username: dosen  
Password: dosen1234  
Login berhasil. Halo, Fachrul Pralienka Bani  
Muhamad, S.ST., M.Kom. (Dosen)
```

Tampilan Login Dosen (Gagal)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen  
(3)Logout: 2  
Username: dosen  
Password: dosen  
Login gagal! Silahkan coba lagi.
```

Tampilan Menu Utama (Mahasiswa)

```
Menu Mahasiswa:  
1. Jadwalkan Sesi  
2. Lihat Jadwal  
3. Tandai Selesai  
4. Ambil Kuis  
5. Lihat Progress  
6. Lihat Riwayat  
7. Logout  
Pilih:
```

Tampilan Menu Utama (Dosen)

```
Menu Dosen:  
1. Tambah Jadwal  
2. Tambah Topik  
3. Buat Kuis  
4. Lihat Semua Jadwal  
5. Logout  
Pilih:
```


Tampilan Menu Mahasiswa (Jadwalkan Sesi Berhasil)

| |
|--|
| ID Jadwal: PBO1 ID Topik: 1 Tanggal (YYYY-MM-DD): 2025-10-22 Durasi (menit): 60 Sesi M1 ditambahkan. |
|--|

Tampilan Menu Dosen (Tambah Jadwal Berhasil)

| |
|--|
| ID Jadwal: PWL1 Nama Jadwal: Pemrograman Web Lanjut Deskripsi: Jadwal mata kuliah ini berisi tentang materi lanjutan mengenai web. Jadwal PWL1 ditambahkan. |
|--|

Tampilan Menu Mahasiswa (Lihat Jadwal Berhasil)

| |
|---|
| Jadwal untuk Alfath Shafira: Sesi: M1 Topik:1 Mahasiswa:Alfath Shafira Tanggal:2025-10-22 Durasi:60 menit Status:Scheduled |
|---|

Tampilan Menu Dosen (Tambah Topik Berhasil)

| |
|--|
| ID Jadwal tujuan: PWL1 ID Topik: 3 Judul Topik: Dasar-dasar Laravel Keterangan: Topik ini mendalami materi tentang menjalankan laravel. Topik 3 ditambahkan ke jadwal PWL1 |
|--|

Tampilan Menu Mahasiswa (Tandai Selesai Berhasil)

| |
|--|
| ID Sesi yang diselesaikan: M1 Sesi M1 ditandai selesai. Total menit: 60 |
|--|

Tampilan Menu Dosen (Buat Kuis Berhasil)

| |
|---|
| ID Topik: 1 ID Kuis: Q3 Berapa banyak soal? 2 Soal 1: Apa kepanjangan OOP? Pilihan 1: Object Oriented Programming Pilihan 2: Oriented Object Programming Pilihan 3: Object Object Programming Pilihan 4: Out Object Programming Jawaban benar (1-4): 1 Soal 2: Dalam bahasa Indonesia, OOP lebih dikenal dengan nama? Pilihan 1: Pemrograman Web Lanjut Pilihan 2: Pemrograman Berorientasi Objek Pilihan 3: Algoritma dan Pemrograman Pilihan 4: Sistem Operasi Jawaban benar (1-4): 2 Kuis Q3 dibuat untuk topik 1 |
|---|

Tampilan Menu Mahasiswa (Ambil Kuis Berhasil)

Tampilan Menu Dosen (Lihat Semua Jadwal Berhasil)

| | |
|---|---|
| <p>ID Kuis: Q3</p> <p>Mulai Kuis Q3</p> <p>Soal: Apa kepanjangan OOP?</p> <p>A) Object Oriented Programming B) Oriented Object Programming C) Object Object Programming D) Out Object Programming</p> <p>Jawab (A-D): A</p> <p>Soal: Dalam bahasa Indonesia, OOP lebih dikenal dengan nama?</p> <p>A) Pemrograman Web Lanjut B) Pemrograman Berorientasi Objek C) Algoritma dan Pemrograman D) Sistem Operasi</p> <p>Jawab (A-D): A</p> <p>Jawaban salah. Jawaban yang benar adalah: B</p> <p>Kuis selesai. Terima kasih telah berpartisipasi dalam mengerjakan kuis ini!</p> <p>Nilai Anda: 50</p> | <p>Jadwal: PBO1 - Pemrograman Berorientasi Objek : Jadwal belajar dasar Java</p> <p>Topik: 1 - Konsep Dasar Topik: 2 - Empat Pilar OOP</p> <p>Jadwal: PWL1 - Pemrograman Web Lanjut : Jadwal mata kuliah ini berisi tentang materi lanjutan mengenai web.</p> <p>Topik: 3 - Dasar-dasar Laravel</p> |
|---|---|

Tampilan Menu Mahasiswa (Lihat Progress Berhasil)

Tampilan Menu Mahasiswa (Lihat Riwayat Berhasil)

| | |
|---|---|
| <p>Progress Alfath Shafira:</p> <p>Total jam belajar: 1 jam 0 menit</p> <p>Topik selesai: 1 / 3 (33%)</p> <p>Rata-rata nilai kuis: 50.0</p> | <p>Riwayat aktivitas Alfath Shafira:</p> <p>Aktivitas Sesi: M1 Tanggal:2025-10-22</p> <p>HasilKuis H1 Tanggal:2025-10-22</p> <p>Mahasiswa:Alfath Shafira Kuis:Q3 Nilai:50</p> |
|---|---|

Tampilan Menu Mahasiswa (Jadwalkan Sesi Belum Berhasil)

Tampilan Menu Dosen (Tambah Topik Gagal)

| | |
|--|--|
| <p>ID Jadwal: 01</p> <p>ID Topik: PBO1</p> <p>Topik tidak ditemukan.Sesi M1 ditambahkan.</p> | <p>ID Jadwal tujuan: 02</p> <p>Jadwal tidak ditemukan.</p> |
|--|--|

Tampilan Menu Mahasiswa (Tandai Selesai Belum Berhasil)

Tampilan Menu Mahasiswa (Ambil Kuis Belum Berhasil)

| | |
|---|--------------------------------------|
| ID Sesi yang diselesaikan: 01 Sesi tidak ditemukan untuk Anda. | ID Kuis: S1 Kuis tidak ditemukan. |
|---|--------------------------------------|

Tampilan Menu Mahasiswa (Lihat Progress Belum Berhasil)

Tampilan Menu Mahasiswa (Lihat Riwayat Belum Berhasil)

| | |
|--|--|
| Progress Alfath Shafira: Total jam belajar: 0 jam 0 menit Topik selesai: 0 / 2 (0%) Rata-rata nilai kuis: 0.0 | Riwayat aktivitas Alfath Shafira: (Belum ada aktivitas) |
|--|--|

Analisis Class

Identifikasi Class dan Attribute

| No | Class (Kata Benda) | Attribute (Kata Benda) |
|----|--------------------|---|
| 1 | Pengguna | id: String nama: String |
| 2 | Instructor | username: String password: String |
| 3 | Student | username: String password: String totalMenitBelajar: int daftarRiwayat: ArrayList<Aktivitas> |
| 4 | Kursus | id: String nama: String deskripsi: String daftarTopik: ArrayList<Topik> |
| 5 | Topik | id: String judul: String keterangan: String daftarKuis: ArrayList<Kuis> |
| 6 | Kuis | id: String topik: Topik soalList: ArrayList<Soal> |
| 7 | Soal | pertanyaan: String pilihan: ArrayList<String> jawabanBenar: int |
| 8 | SesiBelajar | id: String topik: Topik student: Student |

| | | |
|----|----------------------|---|
| | | tanggal: String |
| | | durasiMenit: int |
| | | status: String |
| 9 | HasilKuis | kuis: Kuis |
| | | student: Student |
| | | nilai: int |
| 10 | Aktivitas (abstract) | id: String |
| | | tanggal: String |
| | | tipe: String |
| 11 | StudyTrackApp | daftarInstructor: ArrayList<Instructor> |
| | | daftarStudent: ArrayList<Student> |
| | | daftarKursus: ArrayList<Kursus> |
| | | daftarSesi: ArrayList<SesiBelajar> |
| | | daftarHasil: ArrayList<HasilKuis> |

Identifikasi Method

| No | Class (Kata Benda) | Method (Kata Kerja) | Parameter (Kata Benda) |
|----|--------------------|------------------------------------|---|
| 1 | Pengguna | lihatProfil(): void | - |
| 2 | Instructor | login(): boolean | username, password |
| 3 | Student | login(): boolean | username, password, menit, riwayat, aktivitas |
| | | tambahMenit(): void | |
| | | tambahRiwayat(): void | |
| | | getRiwayat(): ArrayList<Aktivitas> | |
| 4 | Kursus | tambahTopik(): void | topik |
| | | lihatKursus(): void | |
| 5 | Topik | tambahKuis(): void | kuis |
| 6 | Kuis | tambahSoal(): void | soal |
| 7 | Soal | getPertanyaan(): String | - |
| | | getPilihan(): ArrayList<String> | |
| | | getJawabanBenar(): int | |
| 8 | SesiBelajar | tandaiSelesai(): void | - |
| | | lihat(): void | |
| 9 | HasilKuis | tampilkan(): void | - |
| 10 | Aktivitas | tampilkan(): abstract | - |
| 11 | StudyTrackApp | loginStudent() | sesuai kebutuhan input (Scanner, Student, dsb.) |
| | | loginInstructor() | |
| | | tambahKursus() | |
| | | tambahTopik() | |
| | | buatKuis() | |
| | | jadwalkanSesi() | |
| | | lihatJadwal() | |
| | | tandaiSelesai() | |
| | | ambilKuis() | |
| | | lihatProgress() | |
| | | lihatRiwayat() | |
| | | lihatSemuaKursus() | |

Jenis Relasi dan Alasan

a. Dependency

| No | Class Awal | Class Tujuan | Alasan |
|----|---------------|---------------|--|
| 1 | Main | StudyTrackApp | Main menggunakan StudyTrackApp untuk menjalankan aplikasi |
| 2 | StudyTrackApp | Instructor | StudyTrackApp memanggil method login() milik Instructor |
| 3 | StudyTrackApp | Student | StudyTrackApp memanggil method login() dan tambahRiwayat() milik Student |
| 4 | StudyTrackApp | Kursus | StudyTrackApp membuat dan menambah Kursus baru |
| 5 | StudyTrackApp | SesiBelajar | StudyTrackApp membuat objek SesiBelajar untuk menjadwalkan sesi belajar |
| 6 | StudyTrackApp | HasilKuis | StudyTrackApp membuat objek HasilKuis saat student menyelesaikan kuis |
| 7 | StudyTrackApp | Topik | StudyTrackApp menggunakan Topik pada method tambahTopik() |
| 8 | StudyTrackApp | Kuis | StudyTrackApp menggunakan Kuis pada method buatKuis() |

b. Unidirectional Association

| No | Class Awal | Class Tujuan | Alasan |
|----|-------------|--------------|--|
| 1 | Kuis | Topik | Setiap kuis hanya mengetahui topik yang menaunginya, tetapi topik tidak mengetahui isi soal secara langsung. |
| 2 | SesiBelajar | Student | Sesi belajar hanya mengacu pada student yang menjadwalkannya. |
| 3 | SesiBelajar | Topik | Sesi belajar mengacu pada topik tertentu yang dipelajari. |
| 4 | HasilKuis | Kuis | Hasil kuis hanya mengetahui kuis yang diambil. |
| 5 | HasilKuis | Student | Hasil kuis mengacu pada student yang mengerjakan kuis tersebut. |

c. Bidirectional Association

| No | Class Awal | Class Tujuan | Alasan |
|----|------------|--------------|---|
| 1 | Student | Aktivitas | Student memiliki daftar aktivitas, dan aktivitas seperti HasilKuis juga mengacu kembali ke student. |

d. Aggregation

| No | Class Awal | Class Tujuan | Alasan |
|----|------------|--------------|--|
| 1 | Kursus | Topik | Topik merupakan bagian dari kursus, tetapi dapat berdiri sendiri jika kursus dihapus. |
| 2 | Topik | Kuis | Kuis terhubung dengan topik, namun bisa dipindahkan ke topik lain tanpa menghapus objek topik. |

e. Composition

| No | Class Awal | Class Tujuan | Alasan |
|----|---------------|--------------|--|
| 1 | Kuis | Soal | Soal tidak dapat berdiri sendiri di luar kuis. Jika kuis dihapus, semua soal juga ikut terhapus. |
| 2 | Student | Aktivitas | Riwayat aktivitas milik student akan hilang jika student tersebut dihapus dari sistem. |
| 3 | Kursus | Topik | Topik adalah bagian dari Kursus |
| 4 | Topik | Kuis | Kuis adalah bagian dari Topik |
| 5 | StudyTrackApp | Student | Student bagian dari StudyTrackApp (tersimpan di daftarStudent) |
| 6 | StudyTrackApp | Instructor | Instructor bagian dari StudyTrackApp (tersimpan di daftarInstructor) |
| 5 | StudyTrackApp | Kursus | Kursus bagian dari StudyTrackApp (tersimpan di daftarKursus) |
| 6 | StudyTrackApp | SesiBelajar | StudyTrackApp menyimpan daftar sesi belajar (daftarSesi) |
| 7 | StudyTrackApp | HasilKuis | StudyTrackApp menyimpan daftar hasil kuis (daftarHasil) |

f. Generalization

| No | Class Awal | Class Tujuan | Alasan |
|----|------------|--------------|--|
| 1 | Instructor | Pengguna | Instructor mewarisi atribut dan perilaku umum dari Pengguna. |
| 2 | Student | Pengguna | Student juga mewarisi atribut umum dari Pengguna. |
| 3 | HasilKuis | Aktivitas | HasilKuis adalah bentuk khusus dari Aktivitas dengan detail tambahan seperti nilai dan referensi kuis. |

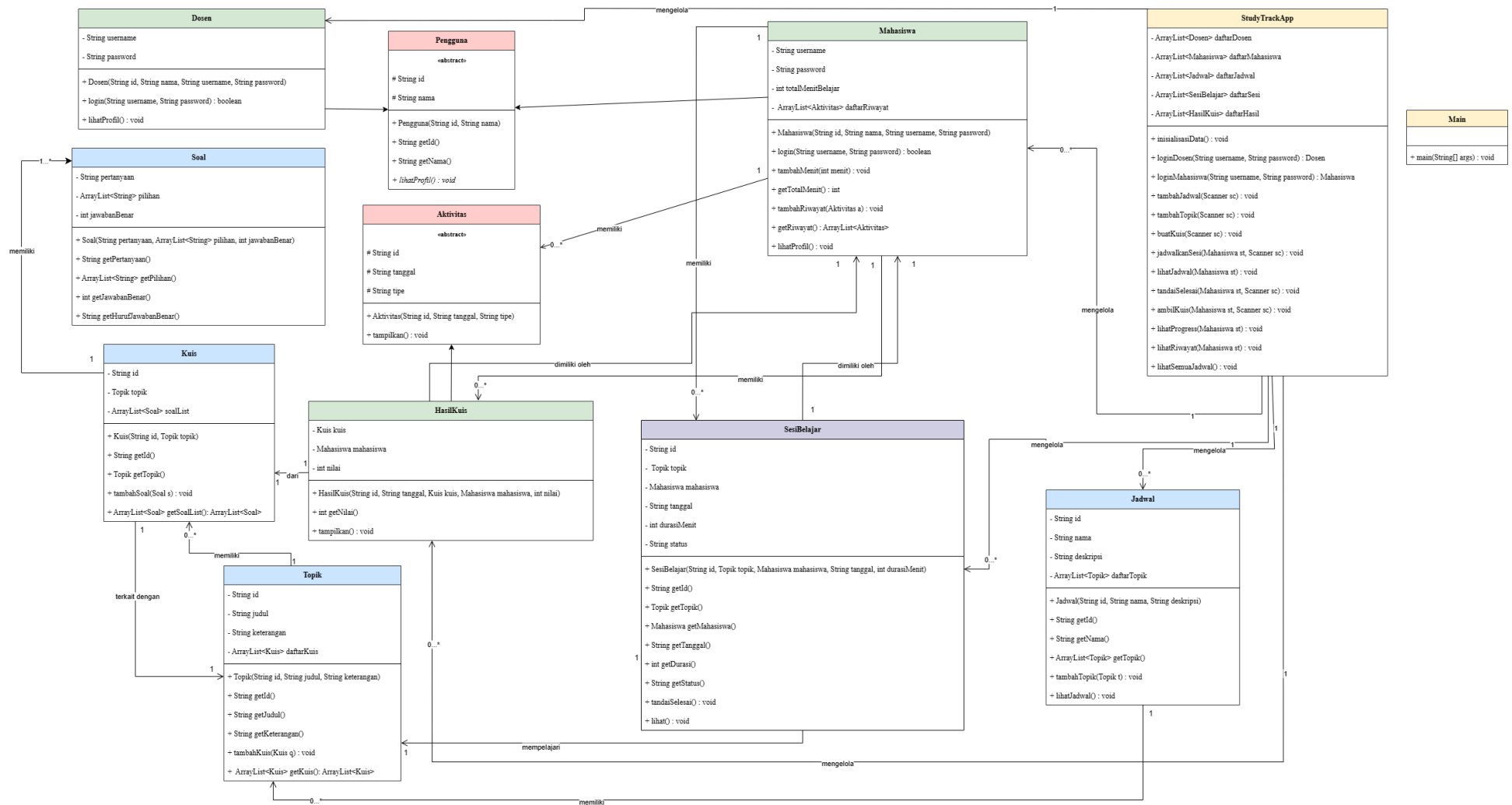
g. Realization

| No | Class Awal | Class Tujuan | Alasan |
|----|------------|--------------|--|
| 1 | - | - | Tidak ada interface yang diimplementasikan dalam sistem ini. |

Multiplicity

| No | Class Awal | Class Tujuan | Relasi | Multiplicity | Keterangan |
|----|---------------|--------------|------------------------|--------------|---|
| 1 | Jadwal | Topik | Composition | 1 & 1...* | Satu jadwal terdiri dari satu atau lebih topik. Jika jadwal dihapus, topik ikut hilang. |
| 2 | Topik | Kuis | Composition | 1 & 1...* | Satu topik memiliki satu atau lebih kuis. Kuis tidak bisa ada tanpa topik. |
| 3 | Kuis | Soal | Composition | 1 & 1...* | Satu <i>kuis</i> terdiri dari satu atau lebih <i>soal</i> . Soal tidak berdiri sendiri tanpa kuis. |
| 4 | Mahasiswa | Aktivitas | Aggregation | 1 & 0...* | Satu <i>mahasiswa</i> dapat memiliki banyak <i>aktivitas</i> (riwayat belajar, hasil kuis, sesi). |
| 5 | SesiBelajar | Mahasiswa | Association | 1 & 1 | Setiap sesi belajar hanya dimiliki oleh satu mahasiswa. |
| 6 | SesiBelajar | Topik | Association | 1 & 1 | <i>Setiap</i> sesi belajar <i>hanya mempelajari satu</i> topik. |
| 7 | HasilKuis | Kuis | Association | 1 & 0...* | Setiap <i>hasil kuis</i> merujuk pada satu <i>kuis</i> tertentu. |
| 8 | Pengguna | Mahasiswa | Generalization | - | Mahasiswa merupakan turunan dari kelas Pengguna |
| 9 | Pengguna | Dosen | Generalization | - | Dosen merupakan turunan dari kelas Pengguna |
| 10 | Aktivitas | HasilKuis | Generalization | - | HasilKuis merupakan turunan (spesialisasi) dari Aktivitas |
| 11 | Mahasiswa | HasilKuis | Association | 1 & 0...* | Mahasiswa dapat memiliki banyak hasil kuis |
| 12 | Dosen | Jadwal | Association | 1 & 0...* | Dosen dapat membuat beberapa Jadwal |
| 13 | StudyTrackApp | | Controller Association | 1 & * | StudyTrackApp berperan sebagai pengendali utama yang mengelola data Dosen, Mahasiswa, Jadwal, Sesi, dan HasilKuis |

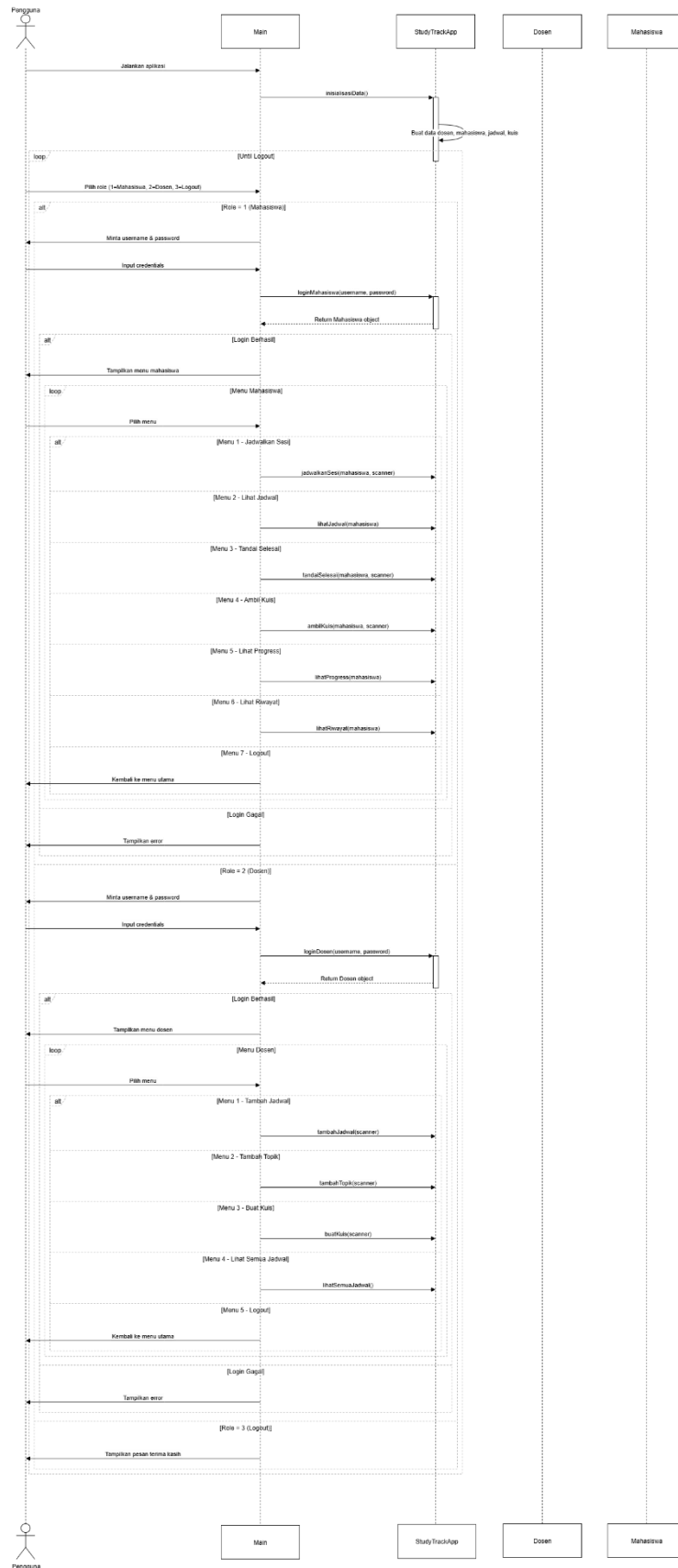
Rancangan UML Class Diagram



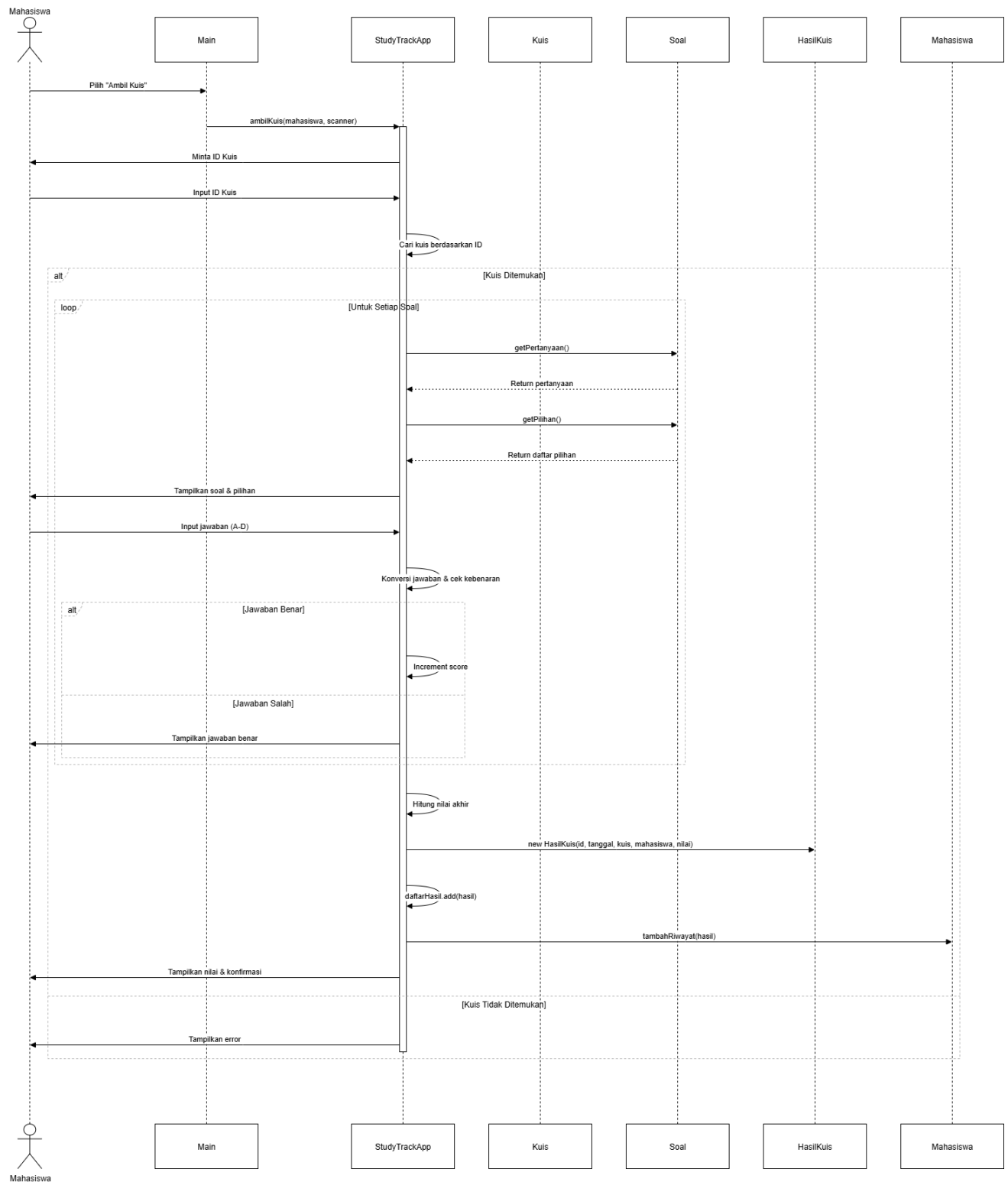
■ Merah Muda: Abstract Classes (Pengguna, Aktivitas).
 ■ Hijau: User Classes (Instructor, Student, HasilKuis).
 ■ Biru: Course Structure (Kursus, Topik, Kuis, Soal).
 ■ Ungu: Learning Session (SesiBelajar).
 ■ Kuning: Application (StudyTrackApp, Main).

Rancangan UML Sequence Diagram

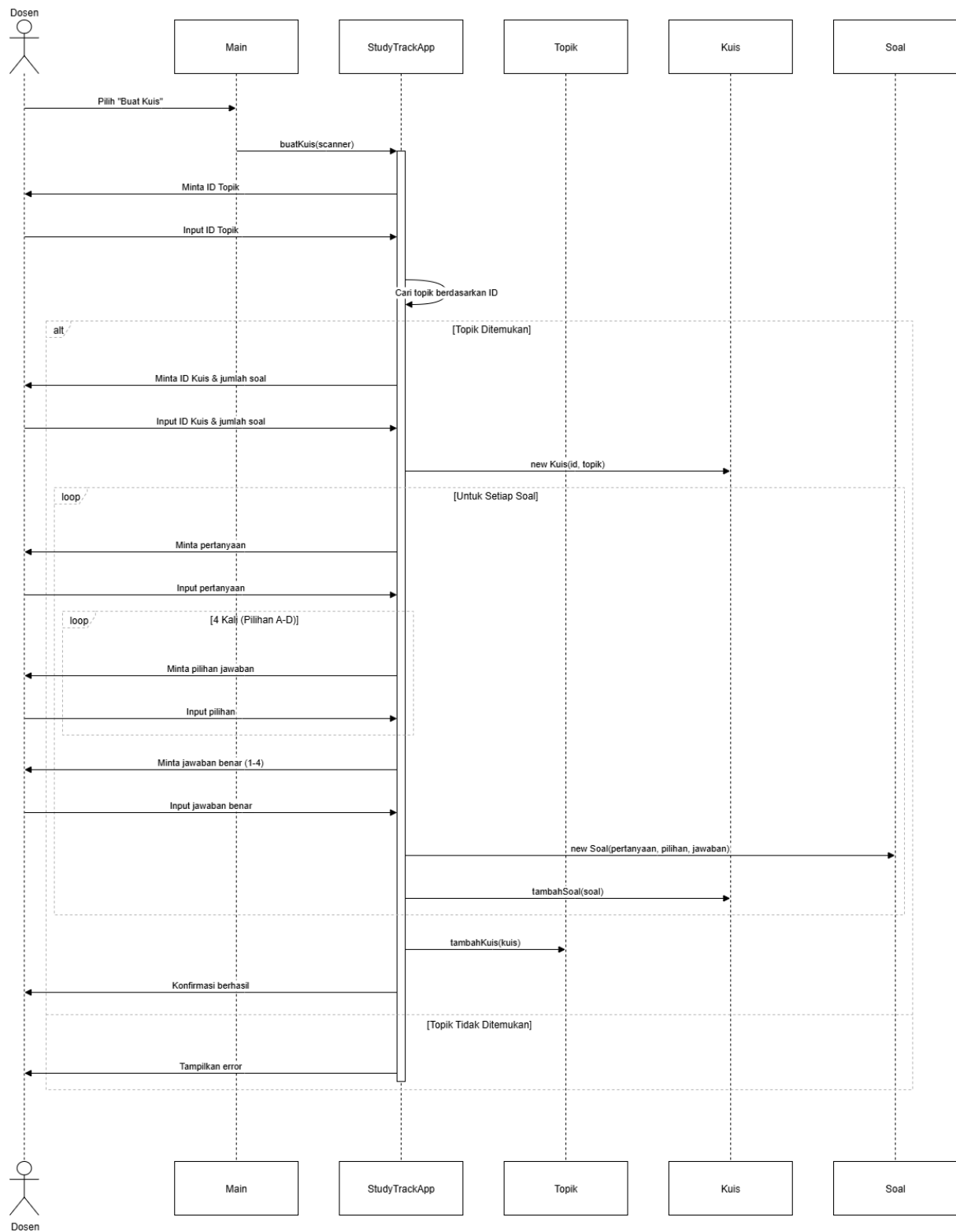
Sequence Diagram: Login



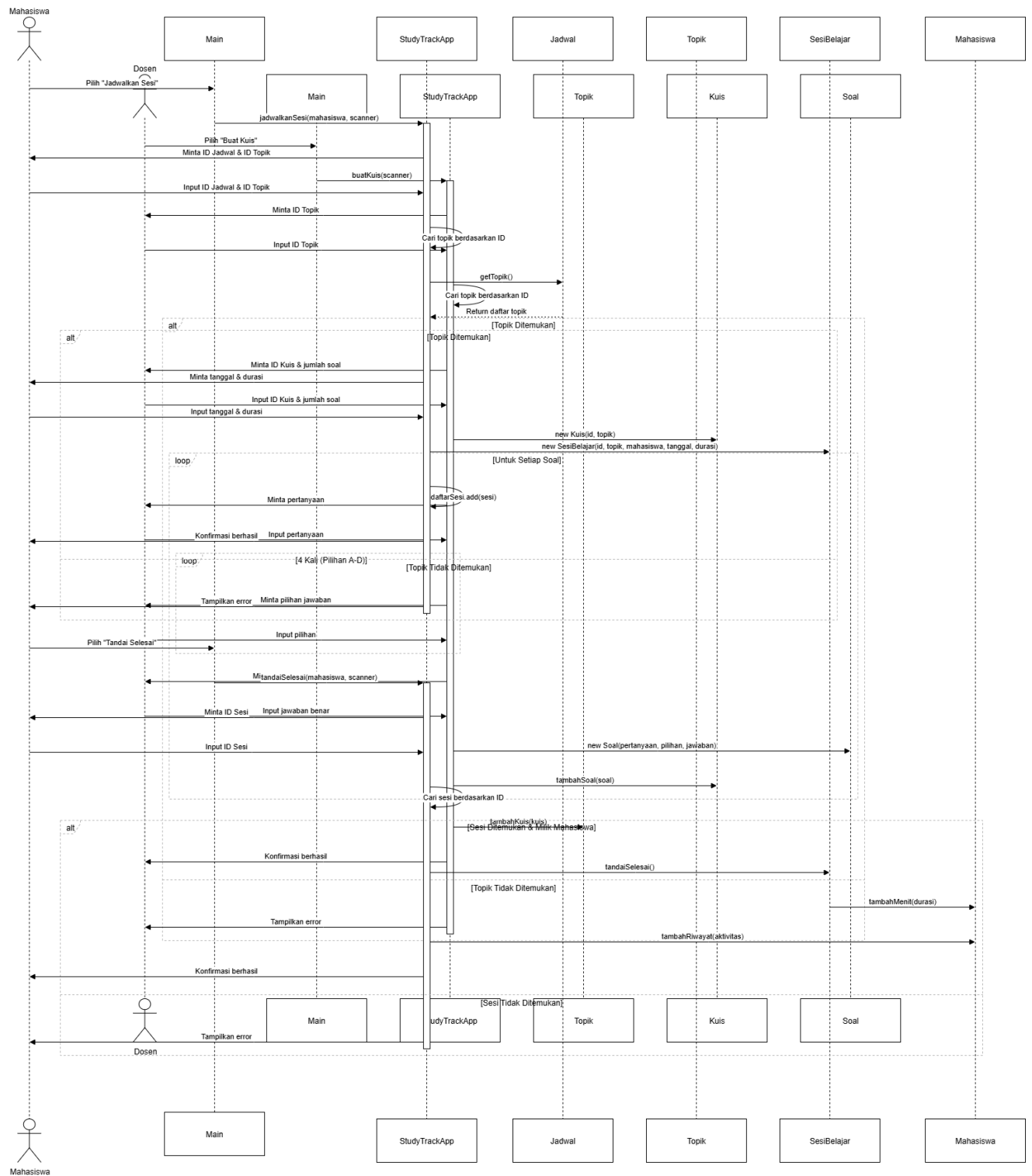
Sequence Diagram: Ambil Kuis



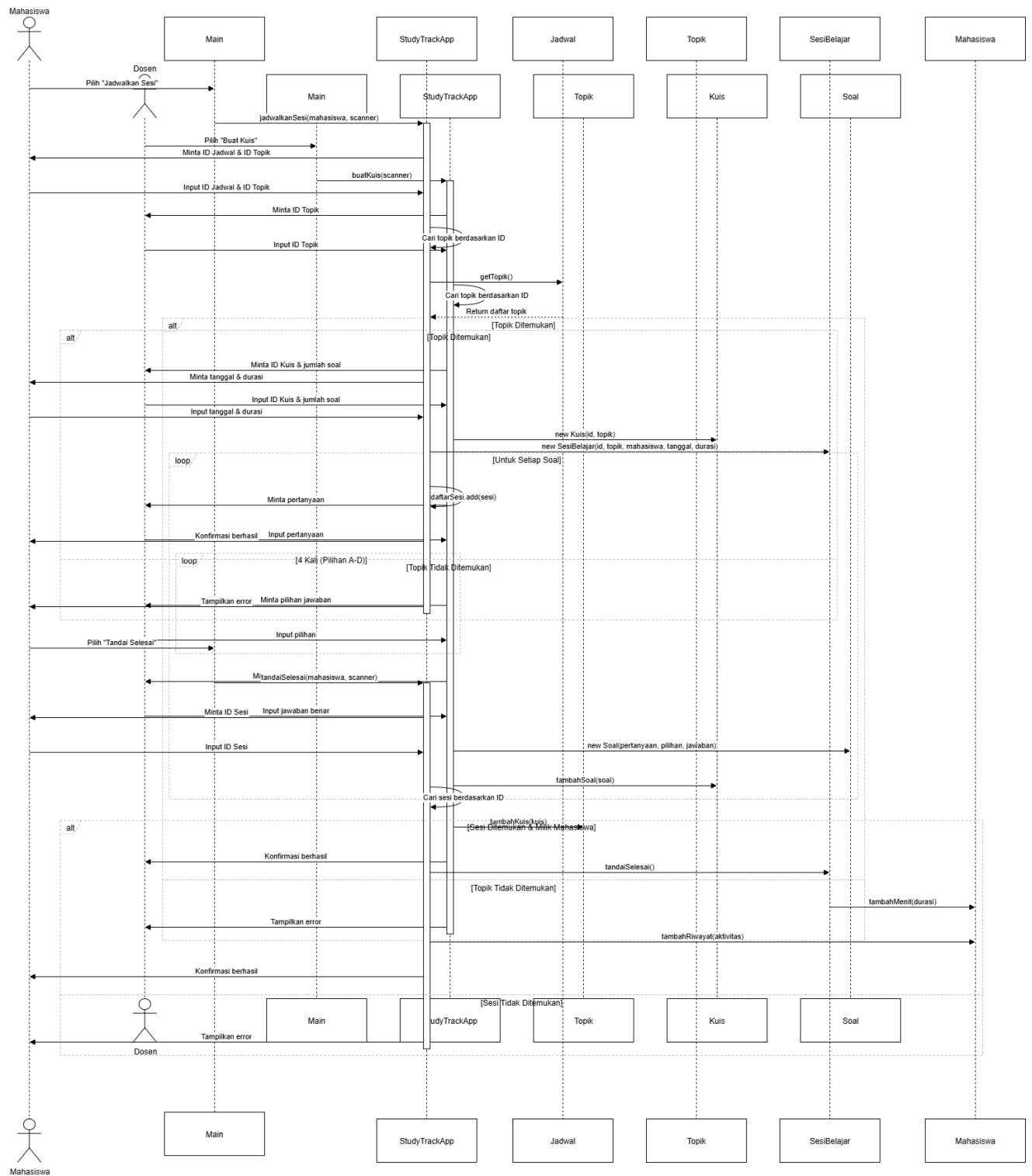
Sequence Diagram: Buat Kuis



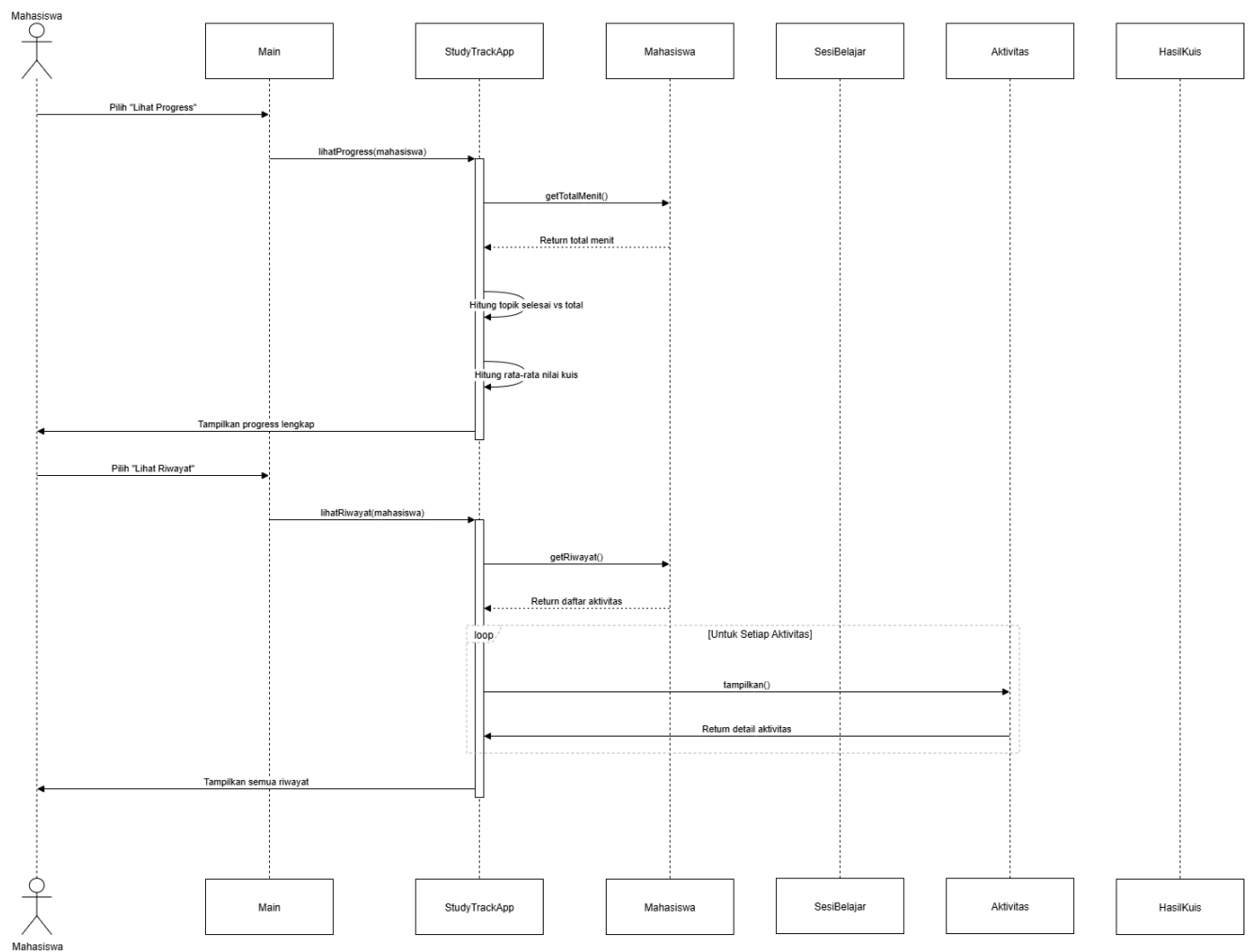
Sequence Diagram: Jadwal Sesi & Tandai Selesai



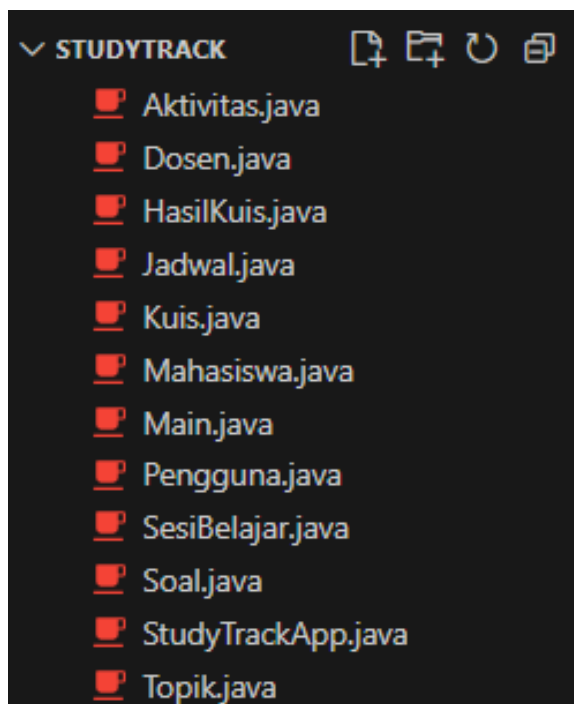
Sequence Diagram: Jadwal Sesi & Tandai Selesai



Sequence Diagram: Progress & Riwayat



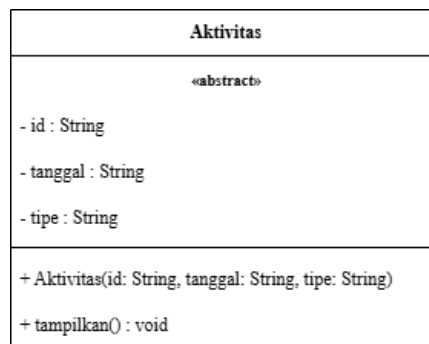
Pra-Code Program



Kode Program

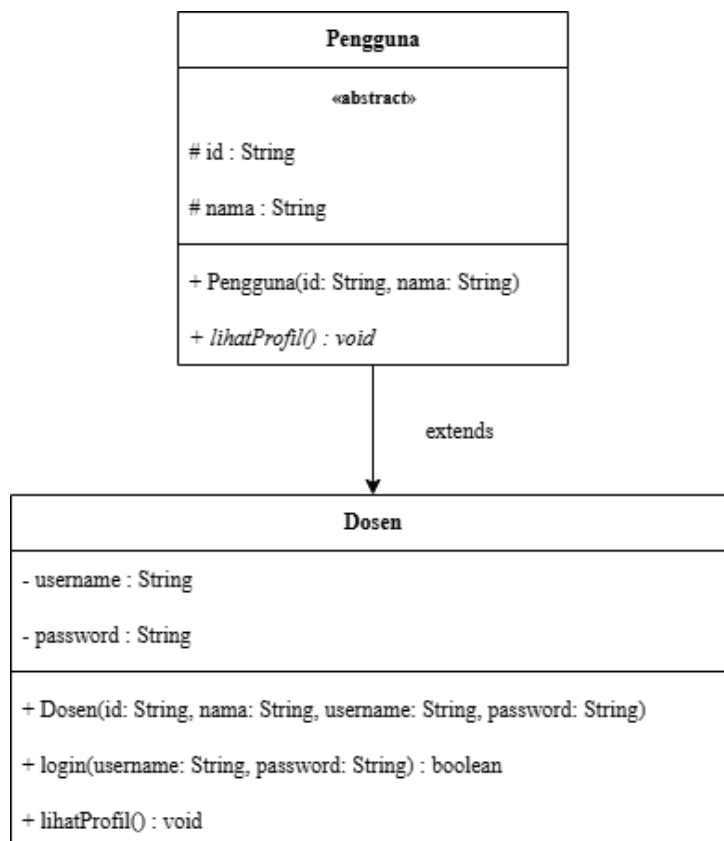
Aktivitas.java

```
public abstract class Aktivitas {  
    protected String id;  
    protected String tanggal;  
    protected String tipe; // "Sesi" atau "Kuis"  
  
    public Aktivitas(String id, String tanggal, String tipe) {  
        this.id = id;  
        this.tanggal = tanggal;  
        this.tipe = tipe;  
    }  
  
    public abstract void tampilkan();  
}
```



Dosen.java

```
public class Dosen extends Pengguna {  
    private String username;  
    private String password;  
  
    public Dosen(String id, String nama, String username, String password) {  
        super(id, nama);  
        this.username = username;  
        this.password = password;  
    }  
  
    public boolean login(String username, String password) {  
        return this.username.equals(username) && this.password.equals(password);  
    }  
  
    @Override  
    public void lihatProfil() {  
        System.out.println("Dosen: " + id + " - " + nama + " (username: " + username + ")");  
    }  
}
```



HasilKuis.java

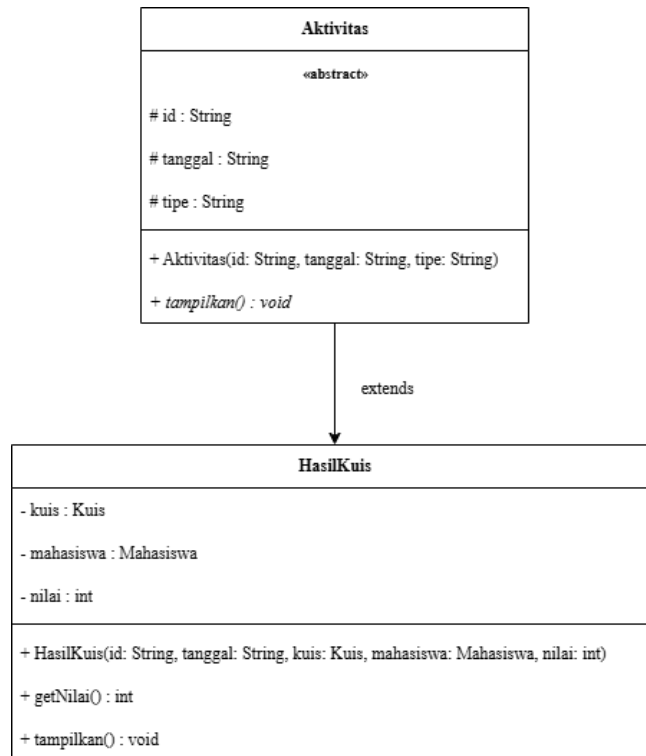
```

public class HasilKuis extends Aktivitas {
    private Kuis kuis;
    private Mahasiswa mahasiswa;
    private int nilai;

    public HasilKuis(String id, String tanggal, Kuis kuis, Mahasiswa mahasiswa, int nilai) {
        super(id, tanggal, "Kuis");
        this.kuis = kuis;
        this.mahasiswa = mahasiswa;
        this.nilai = nilai;
    }

    public int getNilai() { return nilai; }

    @Override
    public void tampilkan() {
        System.out.println("HasilKuis " + id + " Tanggal:" + tanggal + " Mahasiswa:" +
            mahasiswa.getNama() + " Kuis:" + kuis.getId() + " Nilai:" + nilai);
    }
}
  
```

Jadwal.java

```

import java.util.ArrayList;

public class Jadwal {
    private String id;
    private String nama;
    private String deskripsi;
    private ArrayList<Topik> daftarTopik = new ArrayList<>();

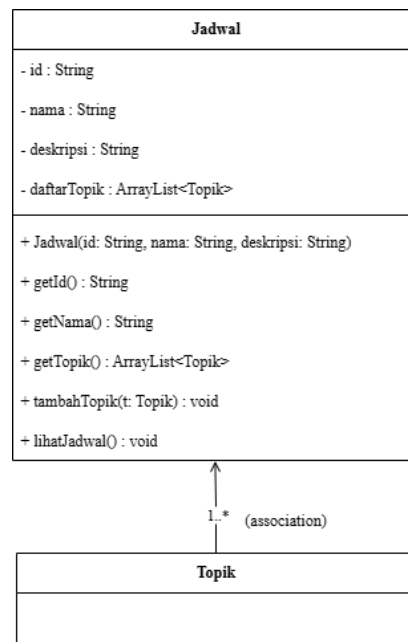
    public Jadwal(String id, String nama, String deskripsi) {
        this.id = id;
        this.nama = nama;
        this.deskripsi = deskripsi;
    }

    public String getId() { return id; }
    public String getNama() { return nama; }
    public ArrayList<Topik> getTopik() { return daftarTopik; }

    public void tambahTopik(Topik t) { daftarTopik.add(t); }

    public void lihatJadwal() {
        System.out.println("Jadwal: " + id + " - " + nama + " : " + deskripsi);
        if (daftarTopik.isEmpty()) System.out.println(" (Belum ada topik)");
        for (Topik t : daftarTopik) System.out.println(" Topik: " + t.getId() + " - " + t.getJudul());
    }
}

```



Kuis.java

```

import java.util.ArrayList;

public class Kuis {
    private String id;
    private Topik topik;
    private ArrayList<Soal> soalList = new ArrayList<>();

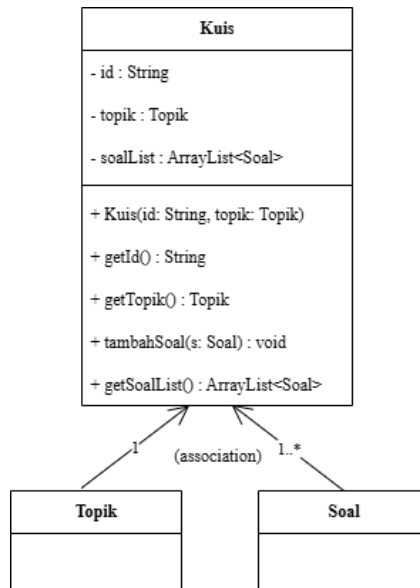
    public Kuis(String id, Topik topik) {
        this.id = id;
        this.topik = topik;
    }

    public String getId() { return id; }
    public Topik getTopik() { return topik; }

    public void tambahSoal(Soal s) { soalList.add(s); }

    public ArrayList<Soal> getSoalList() { return soalList; }
}

```



Mahasiswa.java

```

import java.util.ArrayList;

public class Mahasiswa extends Pengguna {
    private String username;
    private String password;
    private int totalMenitBelajar = 0;
    private ArrayList<Aktivitas> daftarRiwayat = new ArrayList<>();

    public Mahasiswa(String id, String nama, String username, String password) {
        super(id, nama);
        this.username = username;
        this.password = password;
    }

    public boolean login(String username, String password) {
        return this.username.equals(username) && this.password.equals(password);
    }

    public void tambahMenit(int menit) { totalMenitBelajar += menit; }

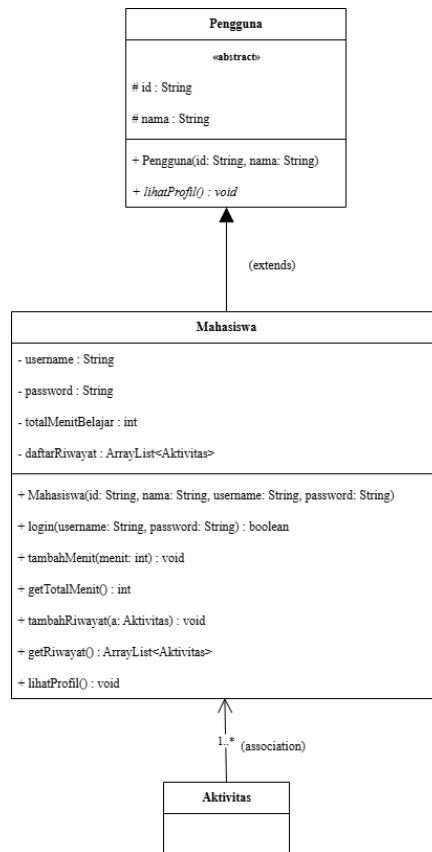
    public int getTotalMenit() { return totalMenitBelajar; }

    public void tambahRiwayat(Aktivitas a) { daftarRiwayat.add(a); }

    public ArrayList<Aktivitas> getRiwayat() { return daftarRiwayat; }

    @Override
    public void lihatProfil() {
        System.out.println("Student: " + id + " - " + nama + " (total menit: " + totalMenitBelajar + ")");
    }
}

```



Main.java

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StudyTrackApp app = new StudyTrackApp();
        app.inisialisasiData();

        boolean exit = false;
        while (!exit) {
            System.out.println("\nSelamat datang di aplikasi StudyTrack!");
            System.out.print("Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: ");
            int role = Integer.parseInt(sc.nextLine());
            if (role == 3) break;

            if (role == 1) {
                System.out.print("Username: "); String u = sc.nextLine();
                System.out.print("Password: "); String p = sc.nextLine();
                Mahasiswa s = app.loginMahasiswa(u,p);
                if (s == null) { System.out.println("Login gagal! Silahkan coba lagi."); continue; }
                System.out.println("Login berhasil. Halo, " + s.getNama() + " (Mahasiswa)");
                boolean keluar = false;
            }
        }
    }
}
  
```

```

while (!keluar) {
    System.out.println("\nMenu Mahasiswa:");
    System.out.println("1. Jadwalkan Sesi");
    System.out.println("2. Lihat Jadwal");
    System.out.println("3. Tandai Selesai");
    System.out.println("4. Ambil Kuis");
    System.out.println("5. Lihat Progress");
    System.out.println("6. Lihat Riwayat");
    System.out.println("7. Logout");
    System.out.print("Pilih menu: ");
    int p1 = Integer.parseInt(sc.nextLine());
    switch (p1) {
        case 1: app.jadwalkanSesi(s, sc); break;
        case 2: app.lihatJadwal(s); break;
        case 3: app.tandaiSelesai(s, sc); break;
        case 4: app.ambilKuis(s, sc); break;
        case 5: app.lihatProgress(s); break;
        case 6: app.lihatRiwayat(s); break;
        case 7: keluar = true; break;
        default: System.out.println("Pilihan tidak valid."); break;
    }
}

} else if (role == 2) {
    System.out.print("Username: "); String u = sc.nextLine();
    System.out.print("Password: "); String p = sc.nextLine();
    Dosen ins = app.loginDosen(u,p);
    if (ins == null) { System.out.println("Login gagal! Silahkan coba lagi."); continue; }
    System.out.println("Login berhasil. Halo, " + ins.getNama() + " (Dosen)");
    boolean keluar = false;
    while (!keluar) {
        System.out.println("\nMenu Dosen:");
        System.out.println("1. Tambah Jadwal");
        System.out.println("2. Tambah Topik");
        System.out.println("3. Buat Kuis");
        System.out.println("4. Lihat Semua Jadwal");
        System.out.println("5. Logout");
        System.out.print("Pilih menu: ");
        int p2 = Integer.parseInt(sc.nextLine());
        switch (p2) {
            case 1: app.tambahJadwal(sc); break;
            case 2: app.tambahTopik(sc); break;
            case 3: app.buatKuis(sc); break;
            case 4: app.lihatSemuaJadwal(); break;
            case 5: keluar = true; break;
            default: System.out.println("Pilihan tidak valid."); break;
        }
    }
}
}

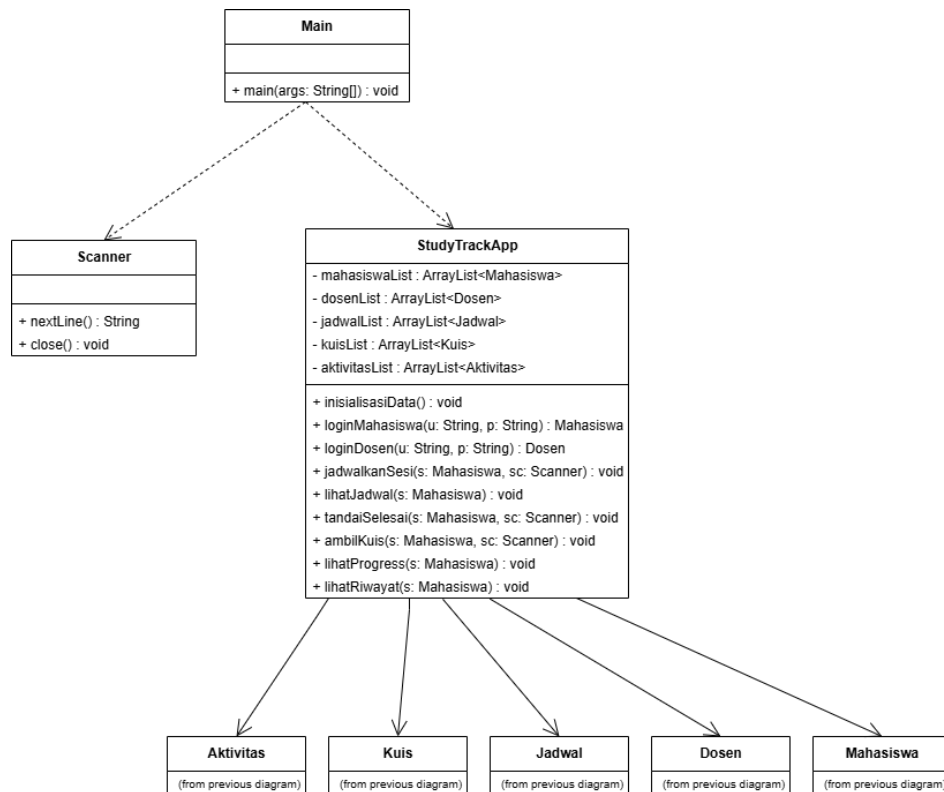
```

```

    }
}

sc.close();
System.out.println("Terima kasih telah menggunakan aplikasi StudyTrack. Sampai jumpa!");
}
}

```



Pengguna.java

```

public abstract class Pengguna {
    protected String id;
    protected String nama;

    public Pengguna(String id, String nama) {
        this.id = id;
        this.nama = nama;
    }

    public String getId() { return id; }
    public String getNama() { return nama; }

    public abstract void lihatProfil();
}

```

| Pengguna |
|--------------------------------------|
| «abstract» |
| # id : String |
| # nama : String |
| + Pengguna(id: String, nama: String) |
| + getId() : String |
| + getNama() : String |
| + lihatProfil() : void |

SesiBelajar.java

```

public class SesiBelajar {
    private String id;
    private Topik topik;
    private Mahasiswa mahasiswa;
    private String tanggal; //YYYY-MM-DD
    private int durasiMenit;
    private String status; //"Scheduled" or "Completed"

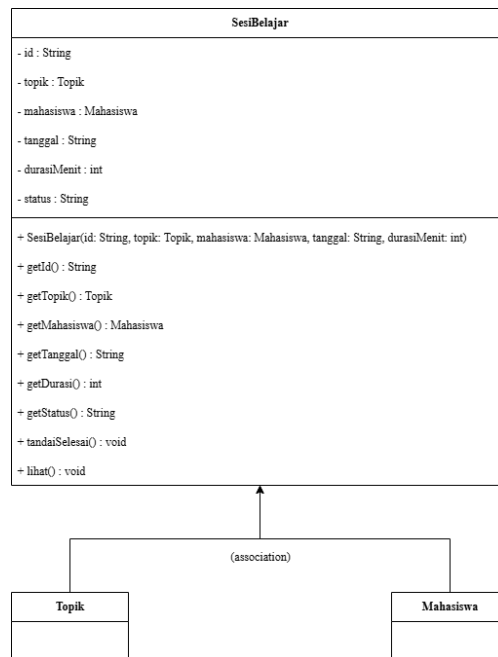
    public SesiBelajar(String id, Topik topik, Mahasiswa mahasiswa, String tanggal, int durasiMenit) {
        this.id = id;
        this.topik = topik;
        this.mahasiswa = mahasiswa;
        this.tanggal = tanggal;
        this.durasiMenit = durasiMenit;
        this.status = "Scheduled";
    }

    public String getId() { return id; }
    public Topik getTopik() { return topik; }
    public Mahasiswa getMahasiswa() { return mahasiswa; }
    public String getTanggal() { return tanggal; }
    public int getDurasi() { return durasiMenit; }
    public String getStatus() { return status; }

    public void tandaiSelesai() {
        if (!status.equals("Completed")) {
            status = "Completed";
            mahasiswa.tambahMenit(durasiMenit);
        }
    }

    public void lihat() {
        System.out.println("Sesi: " + id + " Topik: " + topik.getId() + " Mahasiswa: " + mahasiswa.getNama()
            + " Tanggal: " + tanggal + " Durasi: " + durasiMenit + " menit Status: " + status);
    }
}

```



Soal.java

```

import java.util.ArrayList;

public class Soal {
    private String pertanyaan;
    private ArrayList<String> pilihan;
    private int jawabanBenar; //disimpan dalam bentuk angka (1-4)

    public Soal(String pertanyaan, ArrayList<String> pilihan, int jawabanBenar) {
        this.pertanyaan = pertanyaan;
        this.pilihan = pilihan;
        this.jawabanBenar = jawabanBenar;
    }

    public String getPertanyaan() { return pertanyaan; }
    public ArrayList<String> getPilihan() { return pilihan; }
    public int getJawabanBenar() { return jawabanBenar; }

    //Konversi angka ke huruf (A, B, C, D)
    public String getHurufJawabanBenar() {
        switch (jawabanBenar) {
            case 1: return "A";
            case 2: return "B";
            case 3: return "C";
            case 4: return "D";
            default: return "";
        }
    }
}

```


| Soal |
|--|
| <p>- pertanyaan : String</p> <p>- pilihan : ArrayList<String></p> <p>- jawabanBenar : int</p> |
| <p>+ Soal(pertanyaan: String, pilihan: ArrayList<String>, jawabanBenar: int)</p> <p>+ getPertanyaan() : String</p> <p>+ getPilihan() : ArrayList<String></p> <p>+ getJawabanBenar() : int</p> <p>+ getHurufJawabanBenar() : String</p> |

StudyTrackApp.java

```
import java.util.ArrayList;
import java.util.Scanner;
import java.time.LocalDate;

public class StudyTrackApp {
    private ArrayList<Dosen> daftarDosen = new ArrayList<>();
    private ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();
    private ArrayList<Jadwal> daftarJadwal = new ArrayList<>();
    private ArrayList<SesiBelajar> daftarSesi = new ArrayList<>();
    private ArrayList<HasilKuis> daftarHasil = new ArrayList<>();

    public void inisialisasiData() {
        Dosen ins = new Dosen("D1", "Fachrul Pralienka Bani Muhamad, S.ST.,
M.Kom.", "dosen", "dosen1234");
        daftarDosen.add(ins);
        Mahasiswa s = new Mahasiswa("M1", "Alfath Shafira", "alshfra", "mhs1910");
        daftarMahasiswa.add(s);

        Jadwal k = new Jadwal("PBO1", "Pemrograman Berorientasi Objek", "Jadwal belajar dasar Java");
        Topik t1 = new Topik("1", "Konsep Dasar", "class, object, atribut, method");
        Topik t2 = new Topik("2", "Empat Pilar OOP", "Encapsulation, Inheritance, Polymorphism,
Abstraction");
        k.tambahTopik(t1); k.tambahTopik(t2);
        daftarJadwal.add(k);

        //kuis
        Kuis q = new Kuis("Q1", t1);
        Soal s1 = new Soal("Kapan method overriding terjadi?", new
java.util.ArrayList<String>(java.util.Arrays.asList("Saat dua kelas tidak berhubungan", "Saat subclass
mendefinisikan metode dengan nama dan parameter sama seperti di superclass", "Saat superclass
memiliki lebih dari satu konstruktor", "Saat metode dibuat dengan kata kunci static")), 2);
        q.tambahSoal(s1);
    }
}
```

```

t1.tambahKuis(q);

//kuis
Kuis q2 = new Kuis("Q2", t1);
Soal s2 = new Soal("Apa fungsi utama dari framework web seperti Laravel?", new
java.util.ArrayList<String>(java.util.Arrays.asList("Mengganti HTML dengan bahasa
baru", "Mempermudah pengembangan dengan struktur dan fitur siap pakai", "Menghapus kebutuhan
akan database", "Menjalankan program tanpa server")), 2);
q2.tambahSoal(s2);
t1.tambahKuis(q2);
}

//Login
public Dosen loginDosen(String username, String password) {
    for (Dosen i : daftarDosen) if (i.login(username,password)) return i;
    return null;
}

public Mahasiswa loginMahasiswa(String username, String password) {
    for (Mahasiswa s : daftarMahasiswa) if (s.login(username,password)) return s;
    return null;
}

//Dosen functions
public void tambahJadwal(Scanner sc) {
    System.out.print("ID Jadwal: "); String id = sc.nextLine();
    System.out.print("Nama Jadwal: "); String nama = sc.nextLine();
    System.out.print("Deskripsi: "); String desc = sc.nextLine();
    Jadwal k = new Jadwal(id,nama,desc);
    daftarJadwal.add(k);
    System.out.println("Jadwal " + id + " ditambahkan.");
}

public void tambahTopik(Scanner sc) {
    System.out.print("ID Jadwal tujuan: "); String idk = sc.nextLine();
    Jadwal k = null; for (Jadwal x : daftarJadwal) if (x.getId().equals(idk)) { k = x; break; }
    if (k==null) { System.out.println("Jadwal tidak ditemukan."); return; }
    System.out.print("ID Topik: "); String idt = sc.nextLine();
    System.out.print("Judul Topik: "); String jud = sc.nextLine();
    System.out.print("Keterangan: "); String ket = sc.nextLine();
    Topik t = new Topik(idt,jud,ket);
    k.tambahTopik(t);
    System.out.println("Topik " + idt + " ditambahkan ke jadwal " + idk);
}

public void buatKuis(Scanner sc) {
    System.out.print("ID Topik: "); String idt = sc.nextLine();
    Topik t = null;
    for (Jadwal k : daftarJadwal) for (Topik tp : k.getTopik()) if (tp.getId().equals(idt)) t = tp;

```

```

if (t==null) { System.out.println("Topik tidak ditemukan."); return; }
System.out.print("ID Kuis: "); String idq = sc.nextLine();
Kuis q = new Kuis(idq,t);
System.out.print("Berapa banyak soal? "); int n = Integer.parseInt(sc.nextLine());
for (int i=0;i<n;i++) {
    System.out.print("Soal " + (i+1) + ": "); String pert = sc.nextLine();
    ArrayList<String> pilihan = new ArrayList<>();
    for (int j=1;j<=4;j++) {
        System.out.print("Pilihan " + j + ": "); pilihan.add(sc.nextLine());
    }
    System.out.print("Jawaban benar (1-4): "); int benar = Integer.parseInt(sc.nextLine());
    Soal s = new Soal(pert, pilihan, benar);
    q.tambahSoal(s);
}
t.tambahKuis(q);
System.out.println("Kuis " + idq + " dibuat untuk topik " + idt);
}

//Mahasiswa functions
public void jadwalkanSesi(Mahasiswa st, Scanner sc) {
    System.out.print("ID Jadwal: "); String idk = sc.nextLine();
    System.out.print("ID Topik: "); String idt = sc.nextLine();
    Topik t = null;
    for (Jadwal k : daftarJadwal) if (k.getId().equals(idk)) {
        for (Topik tp : k.getTopik()) if (tp.getId().equals(idt)) t = tp;
    }
    if (t==null) { System.out.println("Topik tidak ditemukan."); return; }
    System.out.print("Tanggal (YYYY-MM-DD): "); String tgl = sc.nextLine();
    System.out.print("Durasi (menit): "); int d = Integer.parseInt(sc.nextLine());
    String idS = "M" + (daftarSesi.size()+1);
    SesiBelajar s = new SesiBelajar(idS, t, st, tgl, d);
    daftarSesi.add(s);
    System.out.println("Sesi " + idS + " ditambahkan.");
}

public void lihatJadwal(Mahasiswa st) {
    System.out.println("Jadwal untuk " + st.getNama() + ".");
    for (SesiBelajar s : daftarSesi) if (s.getMahasiswa().getId().equals(st.getId())) s.lihat();
}

public void tandaiSelesai(Mahasiswa st, Scanner sc) {
    System.out.print("ID Sesi yang diselesaikan: "); String ids = sc.nextLine();
    for (SesiBelajar s : daftarSesi) if (s.getId().equals(ids) &&
s.getMahasiswa().getId().equals(st.getId())) {
        if (s.getStatus().equals("Completed")) { System.out.println("Sesi sudah ditandai selesai.");
return; }
        s.tandaiSelesai();
        // tambahkan aktivitas sesi ke riwayat (simple output)

```

```

        Aktivitas a = new Aktivitas("A" + (st.getRiwayat().size() + 1), LocalDate.now().toString(), "Sesi") {
            @Override public void tampilkan() {
                System.out.println("Aktivitas Sesi: " + ids + " Tanggal: " + this.tanggal);
            }
        };
        st.tambahRiwayat(a);
        System.out.println("Sesi " + ids + " ditandai selesai. Total menit: " + st.getTotalMenit());
        return;
    }
    System.out.println("Sesi tidak ditemukan untuk Anda.");
}

public void ambilKuis(Mahasiswa st, Scanner sc) {
    System.out.print("ID Kuis: ");
    String idq = sc.nextLine();
    Kuis q = null;

    // Cari kuis berdasarkan ID
    for (Jadwal k : daftarJadwal)
        for (Topik t : k.getTopik())
            for (Kuis ku : t.getKuis())
                if (ku.getId().equals(idq))
                    q = ku;

    if (q == null) {
        System.out.println("Kuis tidak ditemukan.");
        return;
    }

    int benar = 0;
    System.out.println("\n Mulai Kuis " + q.getId() + " ");

    //Tampilkan soal satu per satu
    for (Soal s : q.getSoalList()) {
        System.out.println("\nSoal: " + s.getPertanyaan());
        char opsi = 'A';
        for (String p : s.getPilihan()) {
            System.out.println(opsi + " " + p);
            opsi++;
        }

        //Input jawaban dalam huruf
        System.out.print("Jawab (A-D): ");
        char jawChar = sc.nextLine().toUpperCase().charAt(0);

        //Konversi huruf ke angka (A=1, B=2, dst)
        int jaw = jawChar - 'A' + 1;
    }
}

```

```

        if (jaw == s.getJawabanBenar()) {
            benar++;
        } else {
            System.out.println("Jawaban salah. Jawaban yang benar adalah: " + s.getHurufJawabanBenar());
        }
    }

    int nilai = (int) ((double) benar / q.getSoalList().size() * 100);
    String idH = "H" + (daftarHasil.size() + 1);

    HasilKuis hk = new HasilKuis(idH, java.time.LocalDate.now().toString(), q, st, nilai);
    daftarHasil.add(hk);
    st.tambahRiwayat(hk);

    System.out.println("\nKuis selesai. Terima kasih telah berpartisipasi dalam mengerjakan kuis ini!");
    System.out.println("Nilai Anda: " + nilai);
}

public void lihatProgress(Mahasiswa st) {
    int menit = st.getTotalMenit();
    int jam = menit / 60;
    int sisaMenit = menit % 60;
    // hitung topik selesai = ambil jumlah topik dan cek ada sesi completed untuk topik itu oleh
    student
    int totalTopik = 0;
    int selesai = 0;
    for (Jadwal k : daftarJadwal) {
        for (Topik t : k.getTopik()) {
            totalTopik++;
            // cek apakah ada sesi completed untuk topik t
            boolean comp = false;
            for (SesiBelajar s : daftarSesi) {
                if (s.getMahasiswa().getId().equals(st.getId()) && s.getTopik().getId().equals(t.getId()) &&
                    s.getStatus().equals("Completed")) { comp = true; break; }
            }
            if (comp) selesai++;
        }
    }
    //rata2 kuis
    int sum = 0; int count = 0;
    for (Aktivitas a : st.getRiwayat()) {
        if (a instanceof HasilKuis) { sum += ((HasilKuis)a).getNilai(); count++; }
    }
    double rata = count==0 ? 0.0 : (double)sum / count;
    System.out.println("Progress " + st.getNama() + ":");
    System.out.println("Total jam belajar: " + jam + " jam " + sisaMenit + " menit");
    System.out.println("Topik selesai: " + selesai + " / " + totalTopik + " (" +
        (totalTopik==0?0:(selesai*100/totalTopik)) + "%)");
}

```

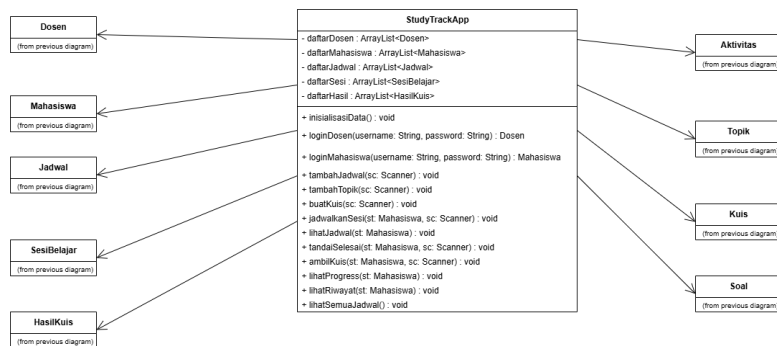
```

        System.out.println("Rata-rata nilai kuis: " + String.format("%.1f", rata));
    }

    public void lihatRiwayat(Mahasiswa st) {
        System.out.println("Riwayat aktivitas " + st.getNama() + ":");
        if (st.getRiwayat().isEmpty()) { System.out.println(" (Belum ada aktivitas)"); return; }
        for (Aktivitas a : st.getRiwayat()) a.tampilkan();
    }

    // Utility: list jadwal
    public void lihatSemuaJadwal() {
        for (Jadwal k : daftarJadwal) k.lihatJadwal();
    }
}

```



Topik.java

```

import java.util.ArrayList;

public class Topik {
    private String id;
    private String judul;
    private String keterangan;
    private ArrayList<Kuis> daftarKuis = new ArrayList<>();

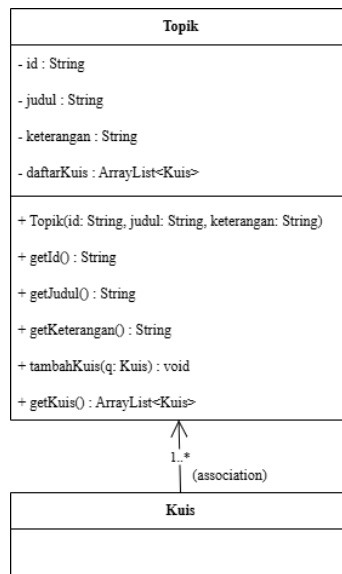
    public Topik(String id, String judul, String keterangan) {
        this.id = id;
        this.judul = judul;
        this.keterangan = keterangan;
    }

    public String getId() { return id; }
    public String getJudul() { return judul; }
    public String getKeterangan() { return keterangan; }

    public void tambahKuis(Kuis q) { daftarKuis.add(q); }

    public ArrayList<Kuis> getKuis() { return daftarKuis; }
}

```



Compile & Run

Compile

```

ALFATH@ALFATH-NOTEBOOK MINGW64 /c/repositorygithub/uts-pbo-study-track-kursus/StudyTrack (main)
$ javac *.java

```

Run

```

ALFATH@ALFATH-NOTEBOOK MINGW64 /c/repositorygithub/uts-pbo-study-track-kursus/StudyTrack (main)
$ java Main

```

```

Selamat datang di aplikasi StudyTrack!
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: █

```

Testing

Skenario Login Mahasiswa Valid (Berhasil)

```

Selamat datang di aplikasi StudyTrack!
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: 1
Username: alshfra
Password: mhs1910
Login berhasil. Halo, Alfath Shafira (Mahasiswa)

Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: █

```

Skenario Login Mahasiswa Tidak Valid (Berhasil)

```

Selamat datang di aplikasi StudyTrack!
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: 1
Username: mahasiswa
Password: mhs
Login gagal! Silahkan coba lagi.

```

Skenario Login Dosen Valid (Berhasil)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: 2  
Username: dosen  
Password: dosen1234  
Login berhasil. Halo, Fachrul Pralienka Bani Muhamad, S.ST., M.Kom. (Dosen)  
  
Menu Dosen:  
1. Tambah Jadwal  
2. Tambah Topik  
3. Buat Kuis  
4. Lihat Semua Jadwal  
5. Logout  
Pilih menu: █
```

Skenario Login Dosen Tidak Valid (Berhasil)

```
Selamat datang di aplikasi StudyTrack!  
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: 2  
Username: dosen  
Password: dosen  
Login gagal! Silahkan coba lagi.
```

Skenario Pilih Menu Mahasiswa Tidak Valid (Berhasil)

```
Menu Mahasiswa:  
1. Jadwalkan Sesi  
2. Lihat Jadwal  
3. Tandai Selesai  
4. Ambil Kuis  
5. Lihat Progress  
6. Lihat Riwayat  
7. Logout  
Pilih menu: 8  
Pilihan tidak valid.
```

Skenario Pilih Menu Dosen Tidak Valid (Berhasil)

```
Menu Dosen:  
1. Tambah Jadwal  
2. Tambah Topik  
3. Buat Kuis  
4. Lihat Semua Jadwal  
5. Logout  
Pilih menu: 6  
Pilihan tidak valid.
```

Skenario Pilih Menu Jadwalkan Sesi Tidak Ditemukan (Berhasil)

```
Menu Mahasiswa:  
1. Jadwalkan Sesi  
2. Lihat Jadwal  
3. Tandai Selesai  
4. Ambil Kuis  
5. Lihat Progress  
6. Lihat Riwayat  
7. Logout  
Pilih menu: 1  
ID Jadwal: 1  
ID Topik: PBO  
Topik tidak ditemukan.
```


Skenario Pilih Menu Jadwalkan Sesi (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 1
ID Jadwal: PB01
ID Topik: 1
Tanggal (YYYY-MM-DD): 2025-10-25
Durasi (menit): 120
Sesi M1 ditambahkan.
```

Skenario Pilih Menu Lihat Jadwal (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 2
Jadwal untuk Alfath Shafira:
Sesi: M1 Topik:1 Mahasiswa:Alfath Shafira Tanggal:2025-10-25 Durasi:120 menit Status:Scheduled
```

Skenario Pilih Menu Tandai Selesai Tidak Ditemukan (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 3
ID Sesi yang diselesaikan: 1
Sesi tidak ditemukan untuk Anda.
```

Skenario Pilih Menu Tandai Selesai (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 3
ID Sesi yang diselesaikan: M1
Sesi M1 ditandai selesai. Total menit: 120
```

Skenario Pilih Menu Ambil Kuis Tidak Ditemukan (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 4
ID Kuis: 1
Kuis tidak ditemukan.
```

Skenario Pilih Menu Ambil Kuis (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 4
ID Kuis: Q1

    Mulai Kuis Q1

Soal: Kapan method overriding terjadi?
A) Saat dua kelas tidak berhubungan
B) Saat subclass mendefinisikan metode dengan nama dan parameter sama seperti di superclass
C) Saat superclass memiliki lebih dari satu konstruktor
D) Saat metode dibuat dengan kata kunci static
Jawab (A-D): C
Jawaban salah. Jawaban yang benar adalah: B

Kuis selesai. Terima kasih telah berpartisipasi dalam mengerjakan kuis ini!
Nilai Anda: 0
```

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 4
ID Kuis: Q1

    Mulai Kuis Q1

Soal: Kapan method overriding terjadi?
A) Saat dua kelas tidak berhubungan
B) Saat subclass mendefinisikan metode dengan nama dan parameter sama seperti di superclass
C) Saat superclass memiliki lebih dari satu konstruktor
D) Saat metode dibuat dengan kata kunci static
Jawab (A-D): B

Kuis selesai. Terima kasih telah berpartisipasi dalam mengerjakan kuis ini!
Nilai Anda: 100
```

Skenario Pilih Menu Lihat Progress (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 5
Progress Alfath Shafira:
Total jam belajar: 2 jam 0 menit
Topik selesai: 1 / 2 (50%)
Rata-rata nilai kuis: 50.0
```

Skenario Pilih Menu Lihat Riwayat (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 6
Riwayat aktivitas Alfath Shafira:
Aktivitas Sesi: M1 Tanggal:2025-10-24
HasilKuis H1 Tanggal:2025-10-24 Mahasiswa:Alfath Shafira Kuis:Q1 Nilai:0
HasilKuis H2 Tanggal:2025-10-24 Mahasiswa:Alfath Shafira Kuis:Q1 Nilai:100
```

Skenario Pilih Menu Lihat Progress Tanpa Data (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 5
Progress Alfath Shafira:
Total jam belajar: 0 jam 0 menit
Topik selesai: 0 / 2 (0%)
Rata-rata nilai kuis: 0.0
```

Skenario Pilih Menu Lihat Riwayat Tanpa Data (Berhasil)

```
Menu Mahasiswa:
1. Jadwalkan Sesi
2. Lihat Jadwal
3. Tandai Selesai
4. Ambil Kuis
5. Lihat Progress
6. Lihat Riwayat
7. Logout
Pilih menu: 6
Riwayat aktivitas Alfath Shafira:
(Belum ada aktivitas)
```

Skenario Pilih Menu Dosen Tambah Topik Tidak Ditemukan (Berhasil)

```
Menu Dosen:
1. Tambah Jadwal
2. Tambah Topik
3. Buat Kuis
4. Lihat Semua Jadwal
5. Logout
Pilih menu: 2
ID Jadwal tujuan: 02
Jadwal tidak ditemukan.
```

Skenario Pilih Menu Dosen Tambah Jadwal (Berhasil)

```
Menu Dosen:
1. Tambah Jadwal
2. Tambah Topik
3. Buat Kuis
4. Lihat Semua Jadwal
5. Logout
Pilih menu: 1
ID Jadwal: PWL1
Nama Jadwal: Pemrograman Web Lanjut
Deskripsi: Jadwal mata kuliah ini berisi tentang materi lanjutan mengenai web.
Jadwal PWL1 ditambahkan.
```

Skenario Pilih Menu Dosen Tambah Topik (Berhasil)

```
Menu Dosen:
1. Tambah Jadwal
2. Tambah Topik
3. Buat Kuis
4. Lihat Semua Jadwal
5. Logout
Pilih menu: 2
ID Jadwal tujuan: PWL1
ID Topik: 3
Judul Topik: Dasar-dasar Laravel
Keterangan: Topik ini mendalami materi tentang menjalankan laravel.
Topik 3 ditambahkan ke jadwal PWL1
```

Skenario Pilih Menu Dosen Buat Kuis (Berhasil)


```
Menu Dosen:
1. Tambah Jadwal
2. Tambah Topik
3. Buat Kuis
4. Lihat Semua Jadwal
5. Logout
Pilih menu: 3
ID Topik: 1
ID Kuis: Q3
Berapa banyak soal? 2
Soal 1: Apa kepanjangan OOP?
Pilihan 1: Object Oriented Programming
Pilihan 2: Oriented Object Programming
Pilihan 3: Object Object Programming
Pilihan 4: Out Object Programming
Jawaban benar (1-4): 1
Soal 2: Dalam bahasa Indonesia, OOP lebih dikenal dengan nama?
Pilihan 1: Pemrograman Web Lanjut
Pilihan 2: Pemrograman Berorientasi Objek
Pilihan 3: Algoritma dan Pemrograman
Pilihan 4: Sistem Operasi
Jawaban benar (1-4): 2
Kuis Q3 dibuat untuk topik 1
```

Skenario Pilih Menu Dosen Lihat Semua Jadwal (Berhasil)

```
Menu Dosen:
1. Tambah Jadwal
2. Tambah Topik
3. Buat Kuis
4. Lihat Semua Jadwal
5. Logout
Pilih menu: 4
Jadwal: PB01 - Pemrograman Berorientasi Objek : Jadwal belajar dasar Java
    Topik: 1 - Konsep Dasar
    Topik: 2 - Empat Pilar OOP
Jadwal: PWL1 - Pemrograman Web Lanjut : Jadwal mata kuliah ini berisi tentang materi lanjutan mengenai web.
    Topik: 3 - Dasar-dasar Laravel
```

Build (Deploy)

```
ALFATH@ALFATH-NOTEBOOK MINGW64 /c/repositorygithub/uts-pbo-study-track-kursus/StudyTrack (main)
$ jar cfe studytrack-app.jar Main *.class
```

 studytrack-app.jar

```
ALFATH@ALFATH-NOTEBOOK MINGW64 /c/repositorygithub/uts-pbo-study-track-kursus/StudyTrack (main)
$ java -jar studytrack-app.jar
```

```
Selamat datang di aplikasi StudyTrack!
Login sebagai (1)Mahasiswa (2)Dosen (3)Logout: █
```

URL Repository

Sajikan alamat URL repositori github yang berisi kode program dan gambar-gambar (png / jpg) yang digunakan pada laporan UTS ini

<https://github.com/ashabyte/uts-pbo-study-track-kursus.git>