

# Perangkat Lunak Penyusunan Jadwal Perkuliahan Menggunakan Algoritma Genetika

<sup>1</sup>Murnawan, <sup>2</sup> Rido Saputra

**Abstract**—Penjadwalan sangat membantu seseorang atau organisasi ketika mengatur dan manajemen waktu, sehingga hal-hal yang sedang dan akan dikerjakan dapat terlaksana sesuai rencana. Dengan hasil jadwal yang optimal akan memberi dampak yang positif untuk kemajuan seseorang atau organisasi dilihat dari segi kedisiplinan dan pemanfaatan waktu.

Komputer adalah sebuah alat yang sempurna untuk mengoptimalkan sesuatu hal selama parameter untuk pengoptimalan masih dapat diinput melalui format elektronik. Diharapkan dengan menggunakan algoritma genetik, akan diperoleh penjadwalan yang optimal yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan matakuliah dan dosen pengajar secara keseluruhan, sehingga tidak adanya permasalahan bentrok jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai dengan fasilitas sarana untuk matakuliah yang ada.

**Index Terms**—Penjadwalan, Algoritma Genetika, Optimasi.

## I. PENDAHULUAN

Dalam kehidupan globalisasi sekarang ini, waktu adalah suatu hal yang sangat berharga bagi seseorang untuk disia-siakan. Baik diantara mereka ada yang berperan sebagai mahasiswa, pembisnis, guru, pedagang, karyawan dan lain sebagainya pasti tidak ingin waktu mereka terbuang secara percuma dengan sesuatu hal yang tak berguna. Membagi waktu memang adalah suatu hal yang sangat sulit untuk dilakukan, walaupun semenjak kecil kita sudah terbiasa dengan kehidupan disiplin biasanya dengan berjalannya waktu dan bertambahnya jumlah kegiatan, kemungkinan besar hal-hal yang penting pun pasti akan terabaikan karena kelalaian dalam pembagian waktu.

Begitupula untuk kegiatan kampus-kampus besar mereka benar-benar memperhitungkan pembagian waktu didalam melakukan kegiatan belajar mengajar agar didapat waktu yang berkesesuaian dengan ketersediaan waktu yang telah diambil oleh dosen dan dapat diikuti oleh mahasiswa yang bersangkutan. Penyusunan jadwal kuliah yang tidak sesuai dapat membuat hal-hal yang telah direncanakan gagal untuk mencapai yang diinginkan, hal ini biasanya disebabkan oleh kurangnya ketersediaan dosen, kurangnya ketersediaan ruangan, jumlah mahasiswa yang melebihi kapasitas, kurangnya ketersediaan peralatan mengajar dan hal-hal lain

yang mengharuskan pihak kampus untuk membuat penyusunan daftar jadwal kuliah tersebut harus dibuat dengan sangat teliti.

Hal yang sama terjadi pula diawal semester di kampus PKN dan STMIK LPKIA. Kegiatan menyusun jadwal kuliah merupakan sesuatu hal yang sangat sulit untuk dilakukan dikarenakan oleh faktor seperti harus menunggu pengembalian formulir perpanjangan kontrak kerja dari tiap-tiap dosen, menentukan waktu dari tiap-tiap dosen untuk mengajar sesuai dengan daftar ketersediaan waktu mengajar

mereka, pengaturan kelas yang sesuai dengan mata pelajaran yang bersangkutan dan ditambah dengan jumlah kelas yang begitu terbatas sehingga proses ini dapat memakan waktu hingga 2-3 minggu. Proses pembentukan jadwal tersebut dilakukan secara berulang setiap awal semester untuk disesuaikan kembali dengan waktu-waktu ketersediaan mengajar dari tiap-tiap dosen dan sarana kelas yang ada.

Penyusunan jadwal dengan memadukan algoritma genetika merupakan suatu langkah untuk mempermudah didalam pembuatan jadwal. Karena dengan proses ini kita menerapkan ilmu *sign* dan komputer atau dengan kata lain menerapkan *Artificial Intelligence* untuk membuat jadwal kuliah. Sebelumnya hasil yang didapat dari pembentukan jadwal masih belum maksimal dari yang diinginkan, seperti diantaranya masih banyak mahasiswa yang ditempatkan di kelas-kelas yang tidak sesuai dengan spesifikasi pendukung dari matakuliah yang bersangkutan, banyak dosen yang mengakhiri proses belajar mengajar dikarenakan waktu yang diberikan saling bertabrakan dengan jadwal ketersediaan mereka, belum lagi terkadang ada kelas yang memiliki waktu jeda kosong melebihi dari aturan yang ditetapkan oleh kampus PKN dan STMIK LPKIA tetapi dengan menerapkan proses penyusunan jadwal dengan pendekatan algoritma genetika ini masalah-masalah tersebut dapat diminimalisir.

## II. METODE PENELITIAN

Penjadwalan untuk kegiatan belajar mengajar dalam suatu kampus adalah hal yang sangat rumit. Permasalahan yang sering disebut dengan *University Time Tabling Problems* (UTP) [9], selain dilihat dari sisi mahasiswa, juga harus dilihat dari sisi dosen, yaitu kemungkinan-kemungkinan dosen akan mengambil lebih dari satu matakuliah yang ada, sebab ada kemungkinan jumlah matakuliah dan jumlah dosen tidak sebanding, sehingga harus dipikirkan juga solusi agar dosen tidak mengambil dua matakuliah berbeda pada hari dan jam yang sama.

Selain itu, harus dipertimbangkan juga ketersediaan kelas sehingga kegiatan belajar dapat dilaksanakan. Salah satu metode yang dapat digunakan untuk menyelesaikan permasalahan tersebut adalah dengan menggunakan pendekatan algoritma genetik.

### A. Dasar Algoritma Genetika

Pada dasarnya Algoritma Genetika memiliki tujuh komponen, tetapi banyak metode yang bervariasi yang diusulkan pada masing-masing komponen tersebut. Masing-masing metode mempunyai kelebihan dan kekurangan. Suatu metode yang bagus untuk masalah A belum tentu bagus untuk masalah B, atau bahkan tidak bisa digunakan untuk masalah C. Tetapi yang pasti menurut [11] ada komponen-komponen yang harus ada dalam algoritma genetika adalah sebagai berikut :

### Skema Pengkodean

Skema pengkodean atau penentuan model dari sistem buatan dengan mendefinisikan spesies-spesies dengan struktur gen dan kromosom yang ditentukan berdasarkan sifat-sifatnya. Terdapat tiga skema yang paling umum digunakan dalam pengkodean [9], yaitu :

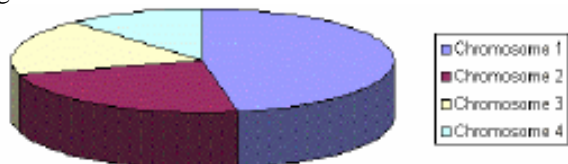
- 1) *Real-number encoding*. Pada skema ini, gen berada dalam interval  $[0, R]$ , dimana  $R$  adalah bilangan real positif dan biasanya  $R=1$ .
- 2) *Discrete decimal encoding*, setiap gen bisa bernilai salah satu bilangan bulat dalam interval  $[0, 9]$ .
- 3) *Binary encoding*, setiap gen hanya bisa bernilai 0 atau 1.

### Nilai Fitness

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Didalam evolusi alam, individu yang bernilai fitness tinggi akan bertahan hidup. Sedangkan individu yang bernilai fitness rendah akan mati atau dengan kata lain perhitungan nilai kebugaran (*Fitness*) dari setiap spesies berdasarkan struktur gennya. Nilai kebugaran ini adalah ukuran optimal tidaknya suatu solusi.

### Seleksi Orang Tua

Pemilihan dua buah kromosom sebagai orang tua, yang akan dipindah-silangkan, biasanya dilakukan secara proporsional sesuai dengan nilai fitnessnya. Pada penelitian ini metode seleksi yang digunakan adalah *roulette-wheel* (roda roulette). Sesuai dengan namanya, metode ini menirukan permainan *roulette-wheel* dimana masing-masing kromosom menempati potongan lingkaran pada roda *roulette* secara proporsional sesuai dengan nilai fitnessnya. Kromosom yang memiliki nilai fitness lebih besar menempati potongan lingkaran yang lebih besar dibanding dengan kromosom bernilai fitness rendah.



Gambar 1. Ilustrasi Roulette Wheel

### Crossover atau Pindah Silang

Salah satu komponen paling penting dalam algoritma genetika adalah *Crossover* atau pindah silang. Sebuah kromosom yang mengarah pada solusi yang bagus bisa diperoleh dari proses memindah-silangkan dua buah kromosom.

*Partial-Mapped Crossover* (PMX) diciptakan oleh Goldberg dan Lingle [11]. PMX merupakan rumusan modifikasi dari pindah silang dua poin. Hal yang penting dari PMX adalah pindah silang 2-poin ditambah dengan beberapa prosedur tambahan.

Langkah-langkahnya sebagai berikut [11]:

- 1) Menentukan 2 posisi pada kromosom dengan aturan acak. Substring yang berada dalam 2 posisi ini dinamakan daerah pemetaan.
- 2) Menukar 2 substring antar induk untuk menghasilkan proto-child.
- 3) Menentukan hubungan pemetaan di antara dua daerah pemetaan.

- 4) Menentukan kromosom keturunan mengacu pada hubungan pemetaan.

### Mutasi

Prosedur mutasi sangatlah sederhana. Untuk semua gen yang ada, jika bilangan random yang dibangkitkan kurang dari probabilitas mutasi yang ditentukan maka ubah gen tersebut menjadi nilai kebalikannya.

### Offspring

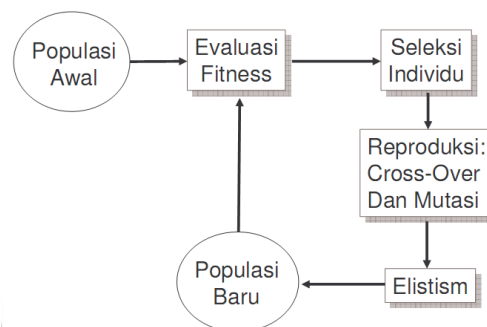
*Offspring* merupakan kromosom baru yang dihasilkan setelah melalui proses-proses di atas. kemudian pada *offspring* tersebut dihitung fitnessnya apakah sudah optimal atau belum, jika sudah optimal berarti *offspring* tersebut merupakan solusi optimal, tetapi jika belum optimal maka akan diseleksi kembali, begitu seterusnya sampai terpenuhi kriteria berhenti. Beberapa kriteria berhenti yang sering digunakan antara lain :

- 1) Berhenti pada generasi tertentu.
- 2) Berhenti setelah dalam beberapa generasi berturut-turut didapatkan nilai fitness tertinggi tidak berubah.
- 3) Berhenti bila dalam  $n$  generasi berikut tidak didapatkan nilai *fitness* yang lebih tinggi

### Proses Elitism

Karena seleksi dilakukan secara random, maka tidak ada jaminan bahwa suatu individu bernilai fitness tertinggi akan selalu terpilih. Kalaupun individu bernilai fitness tinggi terpilih, mungkin saja individu tersebut akan rusak atau nilai fitnessnya menurun karena proses silang. Untuk menjaga agar individu bernilai fitness tinggi tersebut tidak hilang selama evolusi, maka perlu dibuatkan satu atau beberapa kopyannya.

Dengan melakukan proses diatas berulang-ulang, diharapkan induk yang baik akan diperoleh generasi anak dengan spesies-spesies yang lebih baik. Berikut alur dari kerja algoritma secara keseluruhan :



Gambar 2. Ilustrasi Proses Algoritma Genetika

## II. HASIL DAN PEMBAHASAN

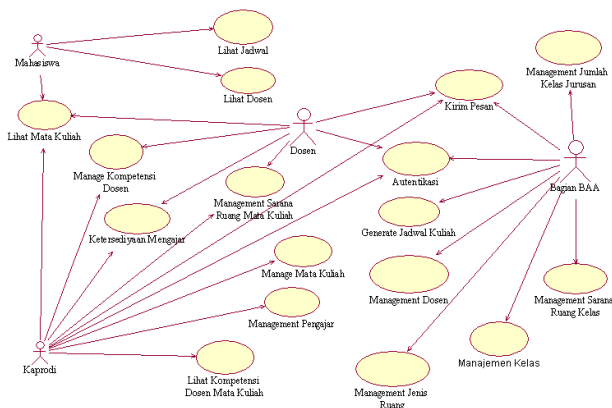
### A. Fungsionalitas

Fungsi dari sebuah sistem dari perspektif para pengguna system dapat di representasikan dengan menggunakan *use case diagram*. *Use case* mendefinisikan apa yang dilakukan oleh sistem dan elemen-elemennya, bukan bagaimana sistem dan elemen-elemennya saling berinteraksi.

*Use case diagram* mengidentifikasi fungsionalitas yang dimiliki oleh sistem (*use case*), pengguna yang berinteraksi dengan sistem dan asosiasi/keterhubungan antara pengguna dengan fungsionalitas sistem.

Berikut ini penjelasan *use case diagram* yang terdapat pada Gambar 1:

- 1) Di dalam *use case diagram* di atas terdapat tiga aktor, yaitu Mahasiswa, Dosen, Kaprodi dan Bagian BAA.
- 2) Mahasiswa adalah pengguna umum atau user yang tidak terdaftar dan hanya bisa melihat hasil akhir dari proses pembuatan jadwal.
- 3) Dosen adalah aktor yang mengisi data ketersediaannya mengajar.
- 4) Dosen harus melakukan autentikasi terlebih dahulu untuk dapat masuk ke sistem.
- 5) Dosen dapat mengisi kompetensinya masing-masing untuk di pertimbangkan oleh aktor kaprodi untuk ditempatkan mengajar matakuliah yang sesuai.
- 6) Kaprodi adalah aktor yang mengatur aktor dosen mana yang dapat memperpanjang kontrak kerjanya.
- 7) Kaprodi harus melakukan autentikasi terlebih dahulu untuk dapat masuk kedalam sistem.
- 8) Kaprodi dapat memanajemen matakuliah mana yang seharusnya dipakai untuk sistem kedepannya.
- 9) Biro Administrasi Akademik (BAA) adalah aktor yang melakukan proses pembentukan jadwal baru sesuai dengan parameter yang telah ditentukan sebelumnya.
- 10) BAA harus melakukan autentikasi terlebih dahulu untuk dapat masuk kedalam sistem.
- 11) Aktor BAA dapat memanajemen seperti sarana ruang, jumlah kelas untuk tiap-tiap jurusan, jenis-jenis ruang yang ada serta jumlah tampung dari tiap-tiap kelas.



Gambar 3. Use Case Diagram Penjadwalan Perkuliahan

## B. Rancangan Class

*Class diagram* adalah sebuah spesifikasi yang jika diinstansiasikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* di bawah ini terdiri dari *class* Lantai, jenisRuangan, saranaRuang, jadwalKuliah, jamKerja, Hari, listKetersediaanMengajar, saranaKelas, Kelas, kriteriaRuangMataKuliah, daftarMataKuliah, programStudi, Kosentrasi, listKelasKosentrasi, masterMataKuliah, kriteriaKompetensiMataKuliah, kompetensiDosen, pesan dan listKompetensiDosen.

*Class* Lantai berfungsi untuk melakukan pengambilan data lantai dari database. Memiliki atribut *idLantai* dan *namaLantai* dan *method* *ambilData*. *Class* Lantai ini berhubungan dengan *class* Kelas.

*Class* jenisRuangan berfungsi untuk melakukan manipulasi data jenisruangan dari database. Memiliki

atribut *idRuangan* dan *namaRuangan* dan *method* *simpan*, *ubah*, *hapus*, *ambilData* dan *jumlahData*. *Class* jenisRuangan ini berhubungan dengan *class* Kelas.

*Class* saranaRuang berfungsi untuk melakukan manipulasi data saranaruangan dari database. Memiliki atribut *idSarana*, *jenis* dan *namaSarana* dan *method* *simpan*, *ubah*, *hapus*, *ambilData* dan *jumlahData*. *Class* saranaRuang ini berhubungan dengan *class* saranaKelas dan kriteriaRuangMataKuliah.

*Class* jadwalKuliah berfungsi untuk melakukan manipulasi data jadwalkuliah dari database. Memiliki atribut *id*, *kodeKelas*, *kodeDosen*, *idMataKuliah*, *idJamKerja*, dan semester serta memiliki *method* *simpan*, *ubah*, *hapus*.

*Class* jamKerja berfungsi untuk melakukan pengambilan data daftarjamkerja dari database. Memiliki atribut *idJamKerja*, *idHari*, *jamAwal*, *jamAkhir* dan opsi serta memiliki *method* *ambilData*. *Class* jamKerja ini berhubungan dengan *class* jadwalKuliah dan listKetersediaanMengajar.

*Class* jenisRuangan berfungsi untuk melakukan manipulasi data ruangan dari database. Memiliki atribut *idRuangan* dan *namaRuangan* dan *method* *simpan*, *ubah*, *hapus*, *ambilData* dan *jumlahData*. *Class* jenisRuangan ini berhubungan dengan *class* Kelas.

*Class* Hari berfungsi untuk melakukan pengambilan data hari dari database. Memiliki atribut *idHari* dan *namaHari* dan *method* *ambilData*. *Class* Hari ini berhubungan dengan *class* jamKerja.

*Class* listKetersediaanMengajar berfungsi untuk melakukan manipulasi data ketersediaanmengajar dari database. Memiliki atribut *id*, *kodeDosen*, *idJamKerja* dan *idMataKuliah* sermta memiliki *method* *simpan*, *hapus* dan *ambilData*.

*Class* programStudi berfungsi untuk melakukan pengambilan data programstudi dari database. Memiliki atribut *idJurusan*, *namaJurusan* dan akreditasi serta memiliki *method* *ambilData*. *Class* programStudi ini berhubungan dengan *class* Kosentrasi.

*Class* Kosentrasi berfungsi untuk melakukan pengambilan data kosentrasi dari database. Memiliki atribut *idKosentrasi*, *namaKosentrasi*, *idJurusan*, *kode* dan jenjang serta memiliki *method* *ambilData*. *Class* Kosentrasi ini berhubungan dengan *class* listKelasKosentrasi dan daftarMataKuliah.

*Class* listKelasKosentrasi berfungsi untuk melakukan manipulasi data listkelaskosentrasi dari database. Memiliki atribut *id*, *idKosentrasi*, semester dan *noUrut* serta memiliki *method* *ambilData*.

*Class* masterMataKuliah berfungsi untuk melakukan manipulasi data matakuliah dari database. Memiliki atribut *idMataKuliah* dan *namaMataKuliah* serta memiliki *method* *simpan*, *ubah*, *hapus*, *ambilData* dan *jumlahData*. *Class* masterMataKuliah ini berhubungan dengan *class* daftarMataKuliah.

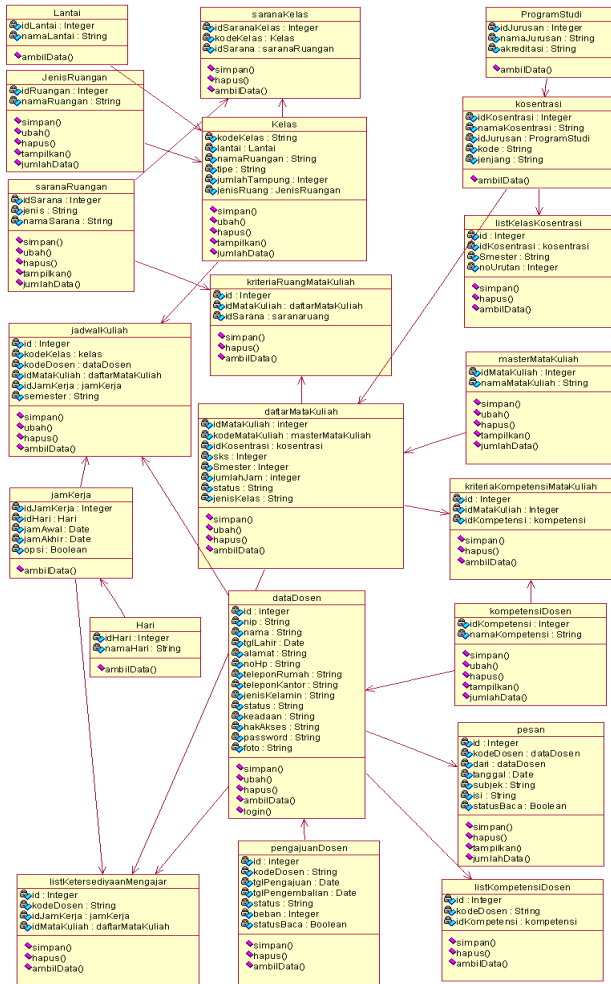
*Class* kriteriaKompetensiMataKuliah berfungsi untuk melakukan manipulasi data kriteriaKompetensiMataKuliah dari database. Memiliki atribut *id*, *idMataKuliah* dan *idKompetensi* serta memiliki *method* *simpan*, *hapus* dan *ambilData*.

*Class* kompetensiDosen berfungsi untuk melakukan manipulasi data kompetensidosen dari database. Memiliki atribut *idKompetensi* dan *namKompetensi* serta memiliki *method* *simpan*, *ubah*, *hapus*, *ambilData* dan *jumlahData*.

Class kompetensiDosen ini berhubungan dengan class dataDosen.

Class pesan berfungsi untuk melakukan manipulasi data pesan dari database. Memiliki atribut id, kodeDosen, dari, tanggal, subjek, isi, dan statusBaca serta memiliki method simpan, hapus, ambilData dan jumlahData.

Class saranaKelas berfungsi untuk melakukan manipulasi data saranakelas dari database. Memiliki atribut idSaranaKelas, kodeKelas dan idsarana serta memiliki method simpan, hapus, ambilData.



Gambar 4. Class Diagram

Class Kelas berfungsi untuk melakukan manipulasi data kelas dari database. Memiliki atribut kodeKelas, lantai, namaRuang, tipe, jumlahTampung dan jenisRuang serta memiliki method simpan, ubah, hapus, ambilData dan jumlahData. Class Kelas ini berhubungan dengan class jadwalKuliah.

Class kriteriaRuangMataKuliah berfungsi untuk melakukan manipulasi data kriteriaruangmatakuliah dari database. Memiliki atribut id, idMataKuliah dan idSarana serta memiliki method simpan, hapus dan ambilData.

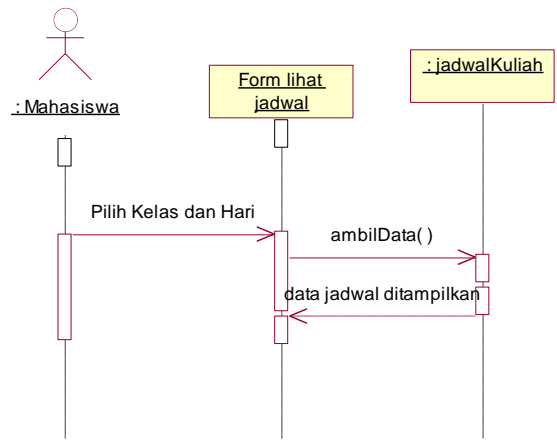
Class daftarMataKuliah berfungsi untuk melakukan manipulasi data daftarMataKuliah dari database. Memiliki atribut idMataKuliah, kodeMataKuliah, idKosentrasi, sks, semester, jumlahJam, status dan jenisKelas serta memiliki method simpan, ubah, hapus dan ambilData. Class daftarMataKuliah ini berhubungan dengan class kriteriaRuangMataKuliah, kriteriaKompetensiMataKuliah dan listKetersediaanMengajar.

Class dataDosen berfungsi untuk melakukan manipulasi data dosen dari database. Memiliki atribut id, nip, nama, tglLahir, alamat, noHp, teleponRumah, teleponKantor, jenisKelamin, status, keadaan, hakAkses, password dan foto serta memiliki method simpan, ubah, hapus dan ambilData dan login. Class dataDosen ini berhubungan dengan class listKompetensiDosen, listKetersediaanMengajar, jadwalKuliah dan Pesan.

Class pengajuanDosen berfungsi untuk melakukan manipulasi data pengajuandosen dari database. Memiliki atribut id, kodeDosen, tglPengajuan, tglPengembalian, status, beban dan statusBaca serta memiliki method simpan, hapus dan ambilData.

C. Sequence Diagram

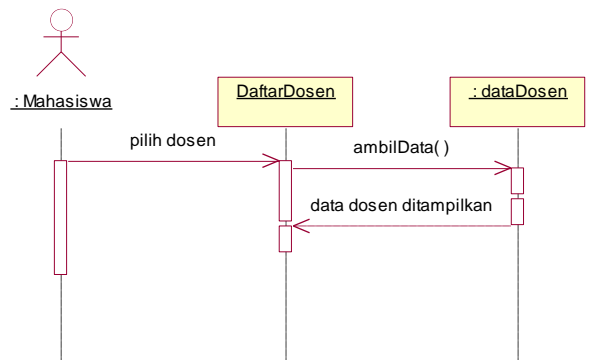
Sequence diagram mendokumentasikan komunikasi atau interaksi antar kelas-kelas. Sequence diagram menunjukkan sejumlah objek dan message (pesan) yang diletakkan diantara objek-objek di dalam use case.



Gambar 5. Sequence Diagram lihat jadwal

Penjelasan sequence diagram lihat jadwal yang terdapat pada Gambar 3:

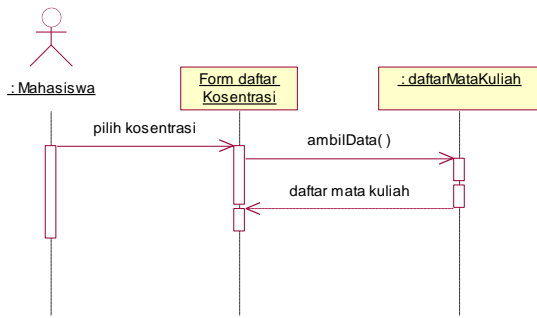
- 1) Mahasiswa atau pengguna umum melakukan lihat jadwal dengan cara masuk ke form lihat jadwal.
- 2) Mahasiswa memilih *combobox* daftar kelas dan hari, jika data sesuai dengan didatabase maka akan tampil daftar jadwal pelajaran.
- 3) Jika tidak ada maka akan tampil pesan kesalahan.



Gambar 6. Sequence Diagram lihat dosen

Penjelasan sequence diagram pada Gambar 4:

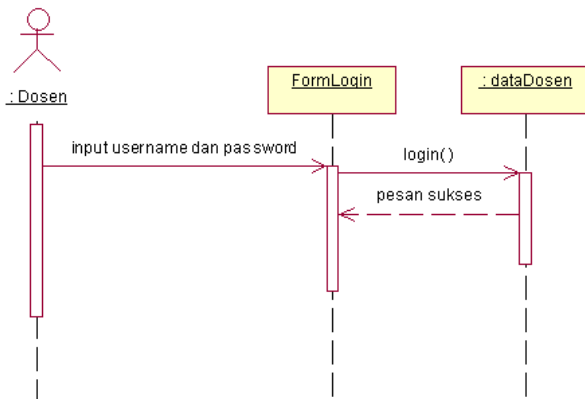
- 1) Mahasiswa atau pengguna umum melakukan lihat dosen dengan cara masuk ke form lihat dosen.
- 2) Mahasiswa mengklik nama dosen.
- 3) Sistem akan menampilkan data dosen yang dipilih.



Gambar 7. Sequence Diagram lihat mataKuliah

Penjelasan *sequence diagram* pada Gambar 5:

- 1) Mahasiswa atau pengguna umum melakukan lihat mataKuliah dengan cara masuk ke form lihat mataKuliah.
- 2) Kemudian mahasiswa memilih jurusan mana yang akan dilihat dengan cara mengklik data jurusan.
- 3) Sistem akan menampilkan daftar makakuliah sesuai dengan pilihan jurusan.



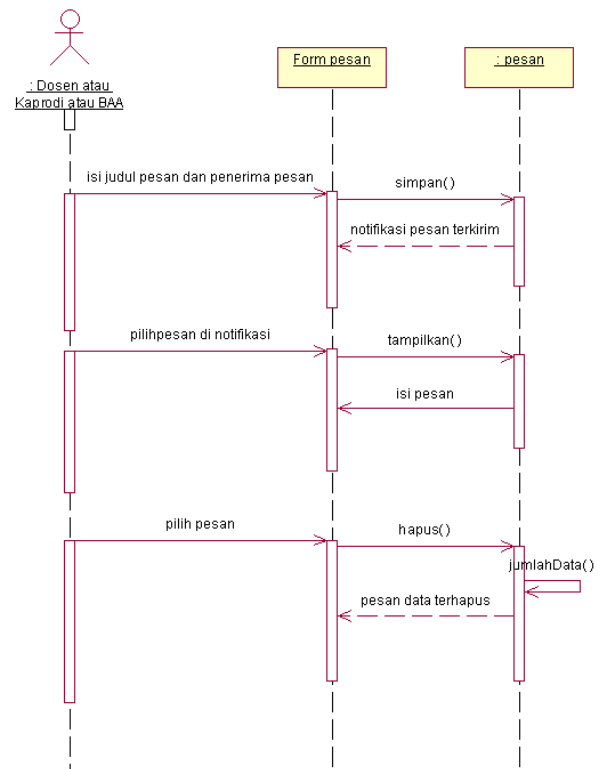
Gambar 8. Sequence Diagram autentikasi

Penjelasan *sequence diagram* pada Gambar 6:

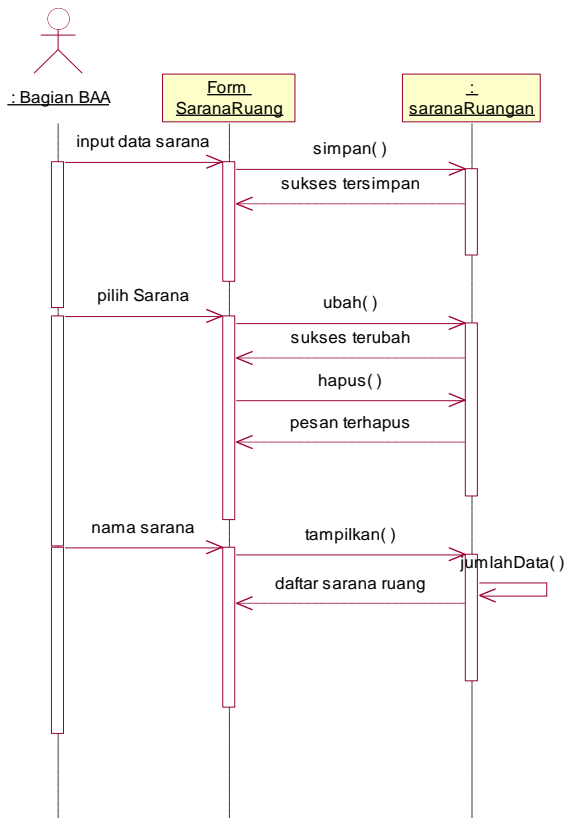
- 1) Dosen, ketua kaprodi atau biro administrasi akademik melakukan login dengan cara masuk ke form login.
- 2) Pengguna memasukkan username dan password.
- 3) Jika data yang di masukkan sesuai dengan yang ada di database, maka pengguna akan di pindahkan ke halaman sesuai dengan hak aksesnya masing-masing.

Berikut ini penjelasan *sequence diagram* pada Gambar 7:

- 1) dosen atau kaprodi atau BAA telah melakukan login terlebih dahulu.
- 2) Untuk mengirim pesan pengguna diharuskan mengisi data penerima pesan, judul pesan dan isi pesan.
- 3) Klik tombol kirim untuk mengirim pesan.
- 4) Jika pengguna akan lihat pesan, langkahnya dengan cara masuk ke form notifikasi pesan masuk.
- 5) Pesan akan ditampilkan oleh sistem.
- 6) Jika ingin menghapus pengguna tinggal memilih menu hapus.



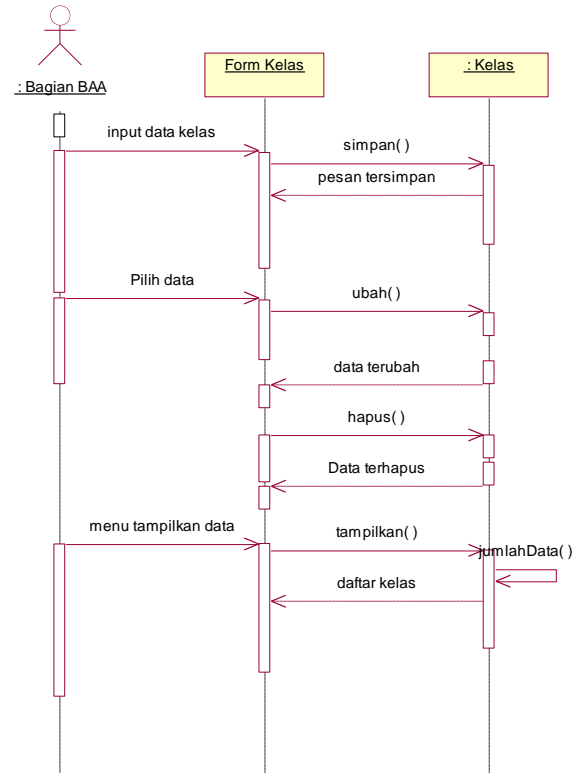
Gambar 9. Sequence Diagram kirim pesan



Gambar 10. Sequence Diagram saranaRuang

Penjelasan *sequence diagram saranaRuang* :

- 1) Biro administrasi Akademik harus sudah login untuk dapat melakukan proses ini.
- 2) Untuk mengisi data, aktor BAA harus mengisi data nama sarana lalu klik simpan untuk proses penyimpanan ke database.
- 3) Untuk mengubah data sarana, aktor BAA tinggal memilih id sarana yang akan diubah lalu isi data yang baru, lalu proses ubah untuk menyimpan perubahan data, jika berhasil data berubah akan tampil pesan data berhasil berubah.
- 4) Untuk proses hapus data pilih idSarana lalu klik menu hapus, jika data terhapus maka akan tampil pesan sukses terhapus.
- 5) Untuk melihat data BAA tinggal klik tombol tampilkan data.



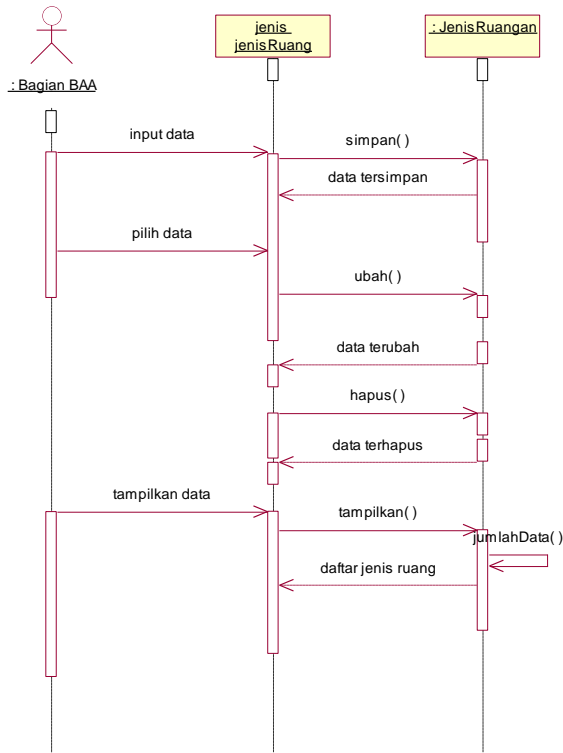
Gambar 11. Sequence Diagram kelas

Penjelasan *sequence diagram kelas* :

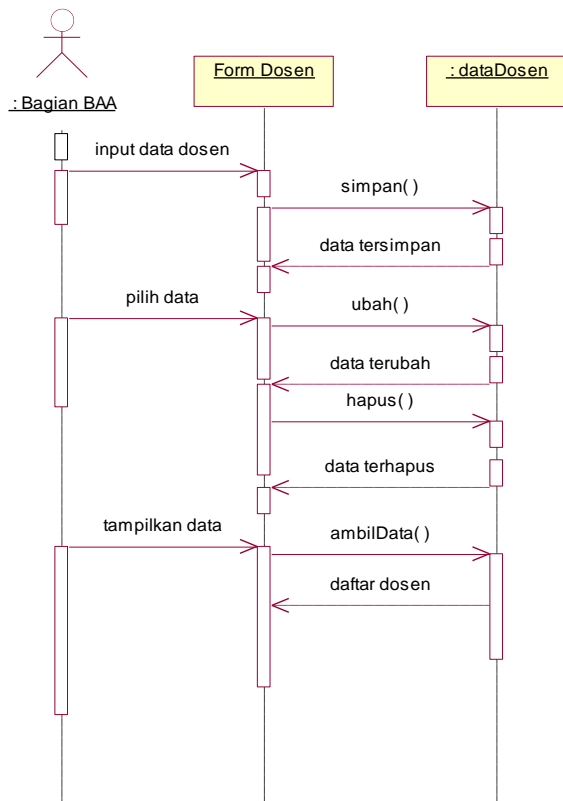
- 1) Aktor BAA sudah melakukan login terlebih dahulu.
- 2) Aktor BAA mengisi data kelas lalu klik simpan untuk menyimpan kedalam database, jika data tersimpan maka akan ada pesan tersimpan kedalam database.
- 3) Untuk mengubah data aktor BAA tinggal memilih data yang akan diubah lalu isi data yang akan diubah lalu klik ubah untuk menyimpan ke database, jika data tersimpan akan tampil pesan berubah.
- 4) Untuk menghapus data BAA tinggal memilih data yang akan dihapus lalu klik menu hapus, jika data terhapus maka akan tampil pesan terhapus.
- 5) Untuk menampilkan data klik menu tampilkan.

Penjelasan *sequence diagram jenisRuang* :

- 1) Aktor BAA harus sudah login untuk bisa melakukan proses ini.
- 2) Aktor BAA mengisi data jenis ruang lalu klik simpan, jika data tersimpan maka sistem akan menampilkan pesan.
- 3) Untuk mengubah data jenis, aktor BAA harus memilih id jenis yang akan diubah lalu klik ubah, jika data berubah maka sistem akan menampilkan pesan data berubah.
- 4) Untuk menampilkan data klik tombol tampilkan data.



Gambar 12. Sequence Diagram jenisRuang

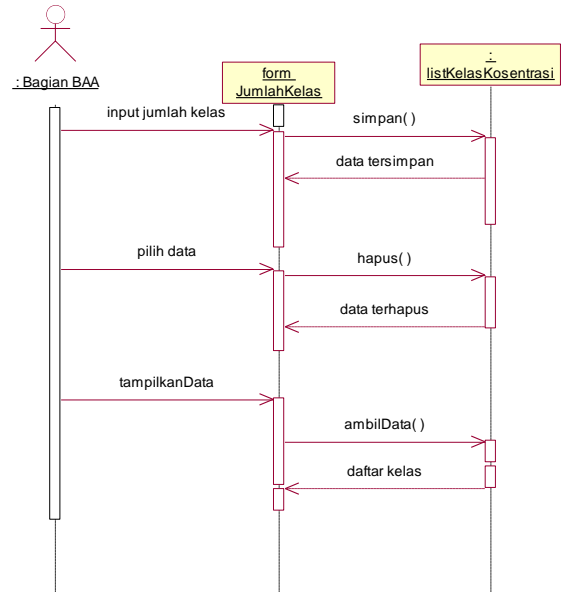


Gambar 13. Sequence Diagram dataDosen

Penjelasan *sequence diagram dataDosen* :

- 1) Aktor BAA sudah melakukan login untuk dapat melakukan proses ini.

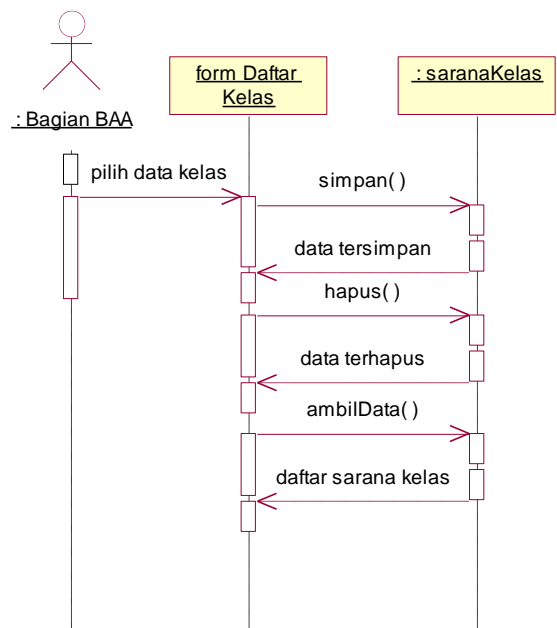
- 2) Aktor BAA mengisi data dosen lalu klik simpan untuk menyimpan kedalam database, jika data tersimpan maka akan tampil pesan sukses penyimpanan.
- 3) Untuk mengubah data BAA harus memilih id dosen lalu mengisi data baru, klik ubah untuk mengubah data. Jika data berubah maka akan tampil pesan sukses berubah.
- 4) Untuk menghapus pilih data yang akan dihapus lalu klik menu hapus. Jika data terhapus maka akan tampil pesan terhapus.
- 5) Untuk menampilkan data klik menu tampilkan data.



Gambar 14. Sequence Diagram jumlahKelas

Penjelasan *sequence diagram jumlahKelas* :

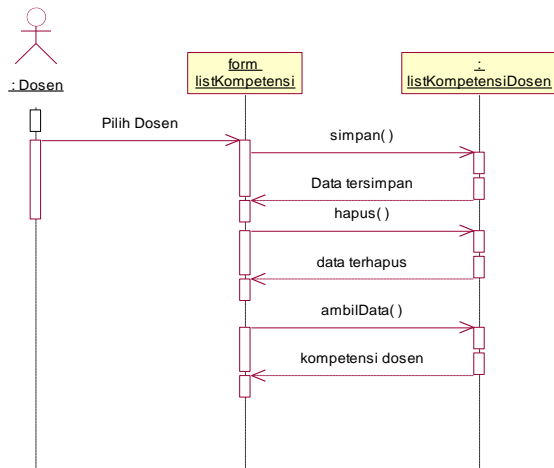
- 1) BAA harus sudah login untuk bisa melakukan proses ini.
- 2) BAA mengisi jumlah data kelas lalu klik simpan untuk menyimpan kedalam database, jika data tersimpan akan ada pesan data tersimpan.
- 3) Untuk hapus data klik id kelas lalu klik hapus.
- 4) Untuk menampilkan data klik menu tampilkan data.



Gambar 15. Sequence Diagram tambahsaranakelas

Penjelasan *sequence diagram* pada Gambar:

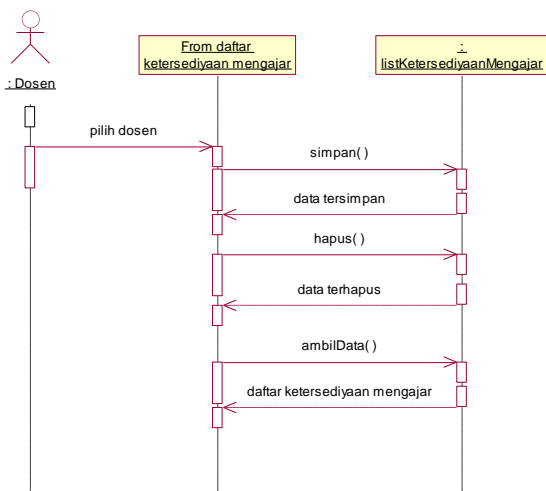
- 1) BAA sudah melakukan login untuk dapat melakukan proses ini.
- 2) Pilih daftar kelas yang akan di tambahkan sarananya, klik simpan untuk menambah data, jika data tersimpan maka akan ada pesan data sukses tersimpan.
- 3) Untuk menghapus pilih id sarana kelas lalu klik menu hapus.
- 4) Jika data terhapus maka akan tampil pesan data terhapus
- 5) Daftar sarana kelas akan ditampilkan dalam bentuk daftar tabel.



Gambar 16. Sequence Diagram daftarKompetensiDosen

Penjelasan *sequence diagram daftarKompetensiDosen* :

- 1) Dosen atau kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) Pilih dosen yang ingin di tambah kompetensinya.
- 3) Klik simpan untuk menyimpan data dalam database, jika data berhasil tersimpan maka akan tampil pesan data tersimpan.
- 4) Untuk menghapus klik menu hapus data, jika data berhasil terhapus maka akan tampil pesan data telah terhapus.
- 5) Daftar data ditampilkan dalam bentuk tabel.

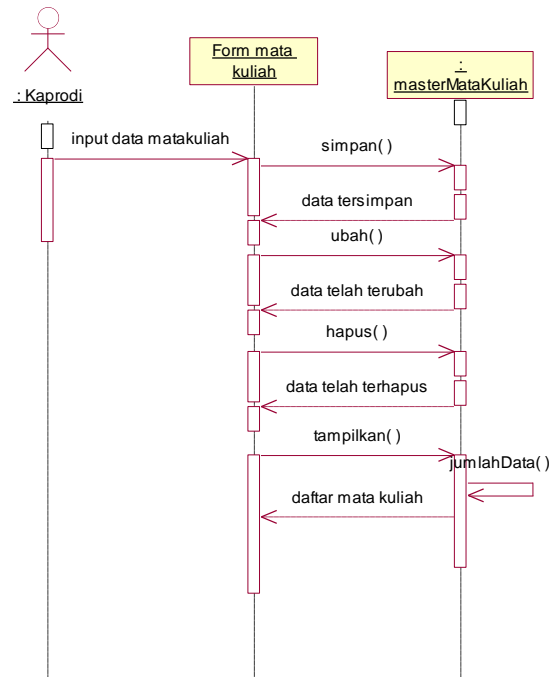


Gambar 17. Sequence Diagram daftarKetersediaanMengajar

Penjelasan *sequence diagram* pada Gambar :

- 1) Dosen atau kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.

- 2) Pilih dosen yang ingin di tambah ketersediaan mengajarnya.
- 3) Klik simpan untuk menyimpan data dalam database, jika data berhasil tersimpan maka akan tampil pesan data tersimpan.
- 4) Untuk menghapus klik menu hapus data, jika data berhasil terhapus maka akan tampil pesan data telah terhapus.
- 5) Daftar data ketersediaan mengajar ditampilkan dalam bentuk tabel.



Gambar 18. Sequence Diagram mastermatakuliah

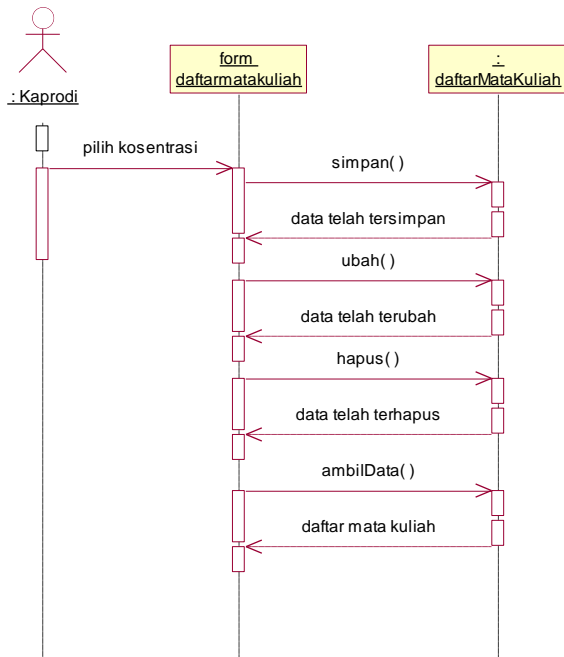
Penjelasan *sequence diagram mastermatakuliah* :

- 1) kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) kaprodi mengisi data matakuliah, klik simpan untuk menyimpan ke database
- 3) Jika data telah tersimpan maka akan ada pesan data telah tersimpan
- 4) Untuk mengubah data klik idmatakuliah lalu klik ubah jika data telah berubah maka akan tampil pesan data telah berubah.
- 5) Untuk menghapus klik menu hapus data.
- 6) Data ditampilkan dalam bentuk tabel.

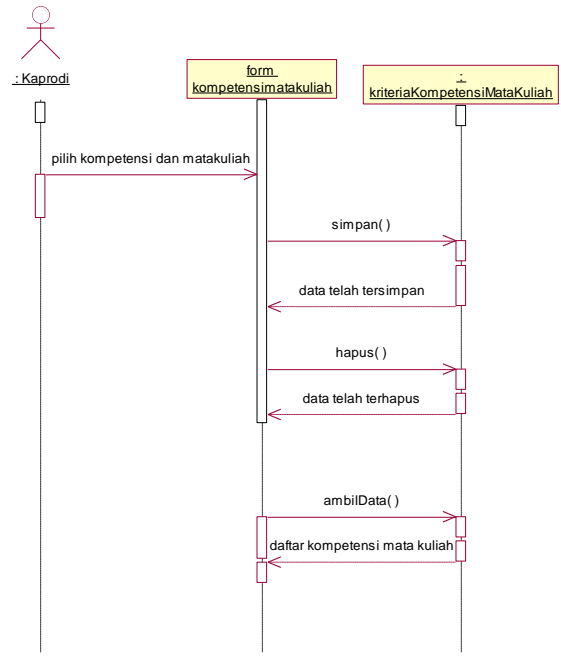
Penjelasan *sequence diagram daftarmatakuliah* :

- 1) kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) kaprodi mengisi data matakuliah berdasarkan jurusan yang terpilih, klik simpan untuk menyimpan ke database
- 3) Jika data telah tersimpan maka akan ada pesan data telah tersimpan
- 4) Untuk mengubah data klik idmatakuliah lalu klik ubah jika data telah berubah maka akan tampil pesan data telah berubah.
- 5) Untuk menghapus klik menu hapus data.
- 6) Data ditampilkan dalam bentuk tabel.

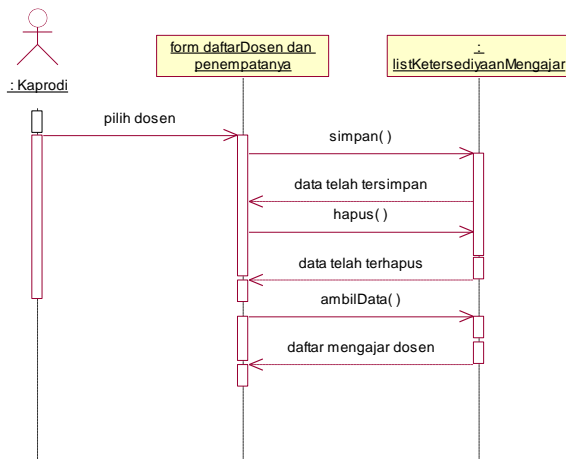




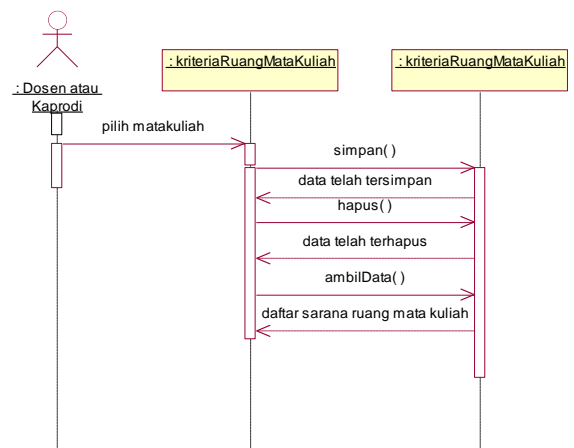
Gambar 19. Sequence Diagram daftarmatakuliah



Gambar 21. Sequence Diagram tambahkompetensimatakuliah



Gambar 20. Sequence Diagram strategipengajar



Gambar 22. Sequence Diagram tambahsaranamatakuliah

Penjelasan *sequence diagram strategipengajar* :

- 1) kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) kaprodi memilih data dosen mana yang mengajar matakuliah yang terpilih berdasarkan data yang dipilih, klik simpan untuk menyimpan ke database
- 3) Jika data telah tersimpan maka akan ada pesan data telah tersimpan
- 4) Untuk menghapus klik menu hapus data.
- 5) Data ditampilkan dalam bentuk tabel.

Penjelasan *sequence diagram tambahkompetensimatakuliah* :

- 1) kaprodi sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) kaprodi memilih data kompetensi mana yang sesuai dengan matakuliah yang terpilih, klik simpan untuk menyimpan ke database
- 3) Jika data telah tersimpan maka akan ada pesan data telah tersimpan
- 4) Untuk menghapus klik menu hapus data.
- 5) Data ditampilkan dalam bentuk tabel.

Penjelasan *sequence diagram tambahsaranamatakuliah* :

- 1) Kaprodi atau Dosen sudah melakukan login terlebih dahulu untuk dapat melakukan proses ini.
- 2) Kaprodi atau Dosen memilih data sarana mana yang sesuai dengan matakuliah yang terpilih, klik simpan untuk menyimpan ke database
- 3) Jika data telah tersimpan maka akan ada pesan data telah tersimpan
- 4) Untuk menghapus klik menu hapus data.
- 5) Data ditampilkan dalam bentuk tabel.

### III KESIMPULAN

Penggunaan algoritma genetic pada proses pembuatan penjadwalan perkuliahan, akan diperoleh penjadwalan yang optimal yaitu kondisi dimana terjadi kombinasi terbaik untuk pasangan matakuliah dan dosen pengajar secara keseluruhan, sehingga tidak adanya permasalahan bentrok jadwal pada sisi mahasiswa, serta ketersediaan ruang yang cukup dan sesuai dengan fasilitas sarana untuk matakuliah yang ada.

Untuk pengembangan selanjutnya sebaiknya algoritma genetika ini di topang oleh algoritma-algoritma lain seperti A\*, heuristic atau dengan metode-metode lain dikarenakan penulis merasa algoritma genetika ini masih kurang tepat didalam menghasilkan *output* jadwal kuliah.

#### DAFTAR PUSTAKA

- [1] Ellen, Randy L. Haupt Sue. *Practical Genetic Algorithms*. Canada: Jhon Wiley & Sons, Inc. Hobuken, new Jersey, 2004.
- [2] Kadir, Abdul. *Dasar Pemograman Web dinamis dengan JSP(Java Server Pages)*. Yogyakarta: Andi Offset, 2004.
- [3] Miftakhul Huda, Cuk Subiyantoro, Bunafit Nugroho. *Membuat aplikasi penjualan dengan java dan mysql*. Jakarta: PT. Elex Media Komputindo, 2008.
- [4] Munawar. *Pemodelan Visual dengan UML*. Yogyakarta: Graha Ilmu, 2005.
- [5] Naba, Dr. Eng. Agus. *Belajar cepat Fuzzy Logic menggunakan MATLAB*. Yogyakarta: C.V ANDI OFFSET, 2009.
- [6] Nugroho, Adi. *Pemograman JAVA menggunakan IDE EWclipse Callisto*. yogyakarta: C.V Andi Offset, 2008.
- [7] Patrick Naughton, Herbert Schildt. *the complete reference Java2*. Berkeley, California: Osborne / McGraw-Hill, 1999.
- [8] Ridwan Sanjaya, SE., S.Kom. *Pengolahan Database Mysql 5 dengan Java 2*. Yogyakarta: Andi Offset, 2005.
- [9] Rothlauf, Franz. *Representations for Genetic and Evolutionari Algorithms*. Netherlands: Springer-Verlag Berlin Heidelberg, 2006.
- [10] Sholiq. *Pemodelan Sistem Informasi dengan UML*. Yogyakarta: Graha Ilmu, 2006.
- [11] Suyanto. *Algoritma Genetika dalam MATLAB*. Yogyakarta: 2005.