

Implementasi Algoritma Round Robin Pada Sistem Penjadwalan Mata Kuliah (Studi Kasus: Universitas Muhammadiyah Bengkulu)

Ardi Wijaya¹, Gunawan,²

¹Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

²Program Studi Sistem Informasi, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

Jl. Bali PO BOX 118. Telp (0736) 227665, Fax (0736) 26161, Bengkulu 38119

E-Mail : ardiwijaya@umb.ac.id¹, gunawan@umb.ac.id²

Abstack: Subjects scheduling is an important routine for teaching and learning process in every semester. These teaching and learning processes involve all students and lecturers, so the arranged subjects scheduling should be able to facilitate the interests both of lecturers and students. If some constraints in scheduling process are not counting well, it will difficult to do subjects scheduling. The constraints are the number of available: subjects, classes, times, and lecturers. The current subjects scheduling process is not efficient. Manual input activity make the scheduling process become complicated and slow. Therefore it need an application which can solve scheduling problems which can be used to arrange the subjects in every departments (majors) in UMB, so that distribution of subjects can be divided equally every day. This problem required an application with a model that can solve scheduling complexities which adjust with the predefined constraints. Round Robin Algorithm is able to make this scheduling process faster and more efficient, so that some items in scheduling process no need to be inserted (input) repeatedly.

Abstrak : Penjadwalan mata kuliah merupakan rutinitas yang sangat penting untuk proses belajar mengajar disetiap semester. Proses belajar mengajar ini melibatkan seluruh mahasiswa dan dosen yang mengajar, sehingga jadwal mata kuliah yang disusun harus dapat memfasilitasi kepentingan dosen dan mahasiswanya. Apabila beberapa batasan dalam proses penjadwalan tidak diperhitungkan dengan baik, maka akan sulit untuk melakukan penjadwalan matakuliah. Batasan– batasan tersebut antara lain matakuliah yang diselenggarakan, kelas yang tersedia, waktu, dan dosen yang mengajar. Proses penentuan jadwal kuliah sekarang ini sering kali tidak efisien. Kegiatan input manual membuat proses penjadwalan menjadi rumit dan lambat. Oleh karena itu dibutuhkan sebuah aplikasi yang dapat memecahkan masalah penjadwalan yang dapat digunakan untuk mengatur jadwal matakuliah pada setiap jurusan di UMB, sehingga distribusi matakuliah tiap harinya dapat dibagi secara merata. Permasalahan ini memerlukan suatu aplikasi dengan suatu model yang mampu mengatasi kerumitan penjadwalan ini yang disesuaikan dengan batasan-batasan yang sudah ditetapkan. Algoritma Round Robin mampu membuat penjadwalan ini lebih cepat dan efisien, sehingga beberapa item pada proses penjadwalan tidak harus (diinputkan) dimasukkan secara berulang-ulang.

Kata kunci : Program Aplikasi, Penjadwalan, Algoritma Round Robin.

I. PENDAHULUAN

Universitas merupakan tempat bagi seseorang yang hendak menimba ilmu, baik itu untuk program diploma, sarjana, magister maupun doctor. Universitas Muhammadiyah Bengkulu (UMB) merupakan salah satu universitas swasta yang berada di Provinsi Bengkulu yang memiliki berbagai macam jurusan.

Penjadwalan mata kuliah merupakan kegiatan yang sangat penting untuk terlaksananya sebuah proses belajar mengajar, tidak hanya bagi dosen yang mengajar, tetapi juga bagi mahasiswa yang mengambil mata kuliah tersebut. Setiap jurusan di Universitas Muhammadiyah Bengkulu memiliki berbagai macam mata kuliah yang harus di ikuti oleh mahasiswanya. Penjadwalan mata kuliah pada sebuah

perguruan tinggi merupakan masalah yang sulit untuk dipecahkan [1]. Tentunya jadwal mata kuliah disetiap jurusan perlu diatur agar tidak bertabrakan dengan jadwal mata kuliah lainnya. Apalagi penentuan jadwal mata kuliah ini juga harus memperhatikan jadwal mengajar dosen yang bersangkutan dan banyaknya jumlah kelas yang ada pada jurusan tersebut.

Banyak kendala yang akan dihadapi ketika membuat penjadwalan mata kuliah. Beberapa batasan yang ada dalam penjadwalan mata kuliah apabila tidak di perhitungkan dengan baik akan menyebabkan sulitnya melakukan penjadwalan yang tepat dan baik. Beberapa kebutuhan dari mahasiswa dan dosen pengampu mata kuliah harus dapat menjadi pertimbangan dalam melakukan penjadwalan [1].

Agar proses penjadwalan mata kuliah dapat tersusun dengan lebih rapi, maka dibutuhkan sebuah algoritma untuk optimasi pembagian jadwal mata kuliah tersebut. Salah satu algoritma optimasi yang digunakan penulis dalam kesempatan ini adalah algoritma *Round Robin*. Penjadwalan *Round Robin* adalah penjadwalan proses menerapkan strategi preemptive, bukan di-preempt oleh proses lain, tapi terutama oleh penjadwal berdasarkan jatah waktu pemroses yang disebut kwanta (quantum)[2].

II. TINJAUAN PUSTAKA

2.1. Pengeritian Implementasi

Menurut Kamus Besar Bahasa Indonesia, implementasi adalah pelaksanaan dan penerapan, dimana kedua hal ini bermaksud untuk mencari bentuk tentang hal yang disepakati terlebih dahulu. Implementasi adalah proses untuk memastikan terlaksananya suatu kebijakan dan tercapainya kebijakan tersebut. Tujuan dari implementasi sebuah sistem ialah untuk menyelesaikan desain sistem yang telah disetujui, menguji serta mendokumentasikan program-program dan prosedur sistem yang diperlukan, memastikan bahwa personil yang terlibat dapat mengoperasikan sistem yang baru dan memastikan bahwa konversi sistem lama ke sistem baru dapat berjalan dengan baik dan benar [3].

2.2. Algoritma

Algoritma adalah urutan logis langkah-langkah penyelesaian masalah. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, dalam hal ini berbeda dengan heuristik. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai [4].

2.2.1. Penjadwalan Proses

Tujuan dari multi-programing adalah untuk mempunyai proses berjalan secara bersamaan, untuk memaksimalkan kinerja dari CPU [5]. Untuk sistem uniprosesor, tidak pernah ada proses yang berjalan lebih dari satu. Bila ada proses yang lebih dari satu maka yang lain harus mengantri sampai bebas.

Tujuan penjadwalan adalah untuk meminilakan total biaya computer dan waktu tunggu user [5]. Dalam prakteknya ada dua masalah yang sering dihadapi, waktu pelayanan terhadap proses dapat dikurangi dengan memperhatikan waktu prosesor yang hilang akibat intervensi user, perangkat keras yang lambat, dan multiplexing sumber daya. Hal ini biasanya memiliki pengaruh yang drastis pada mode operasi ditawarkan pada user [5].

2.3. Algoritma *Round Robin* (RR)

Round Robin adalah penjadwalan proses menerapkan strategi preemptive, bukan di-preempt oleh proses lain, tapi terutama oleh penjadwal berdasarkan jatah waktu pemroses yang disebut kwanta (quantum) (Santika & Hansun, 2014). Algoritma *Round Robin* disebut juga Fair Time Scheduling, memiliki prinsip dasar, yaitu semua sumber antriandianggap sama sehingga diberi waktu yang disebut time quantum. Jika time quantum habis atau proses selesai, maka proses berlanjut ke antrian berikutnya. Penjadwalan ini cukup adil karena tidak ada antrian yang diprioritaskan, semua mendapat jatah waktu yang sama. Secara spesifik, penjadwalan ini akan menjadwalkan user- j pada TTI(Transmission Time Interval) ke- k jika:

$$User-j = \text{mod}((k-1), N) + 1 \quad (1) \quad [4]$$

N adalah jumlah user yang sedang aktif dalam sistem. Dari persamaan tersebut terlihat bahwa teknik penjadwalan iniindependen terhadap kondisi propagasi maupun karakteristik kanal, sehingga tidak menganggap adanya keragamankondisi multi-user.

Algoritma *Round Robin* merupakan salah satu algoritma penjadwalan yang paling tua, sederhana, adil, banyak digunakan algoritmanya dan mudah diimplementasikan. Penjadwalan ini bukan dijalankan oleh proses lain tetapi

oleh penjadwal, berdasarkan lama waktu berjalannya proses (*preempt by time*). Penjadwalan tanpa prioritas berasumsi bahwa semua proses memiliki kepentingan yang sama, sehingga tidak ada prioritas tertentu. Semua proses dianggap penting sehingga diberi sejumlah waktu oleh pemroses yang disebut kwanta (*quantum*) atau *time slice* dimana proses itu berjalan. Jika proses masih *running* sampai akhir quantum, maka CPU akan mem-*preempt* proses itu dan memberikannya ke proses lain. Penjadwal membutuhkannya dengan memelihara daftar proses dari *runnable*. Ketika quantum habis untuk satu proses tertentu, maka proses tersebut akan diletakkan diakhir daftar (*list*).

2.3.1. Ketentuan

Semua proses dianggap penting dan diberi sejumlah waktu proses yang disebut kwanta (*quantum*) atau *time slice* dimana proses itu berjalan. Ketentuan algoritma *Round Robin* adalah sebagai berikut [6]. Jika quantum dan proses belum selesai maka proses menjadi *runnable* dan pemroses dialihkan ke proses lain.

1. Jika kwanta (*quantum*) dan proses belum selesai maka proses akan menjadi *runnable* dan pemroses dialihkan ke proses lain.
2. Jika kwanta (*quantum*) belum habis dan proses menunggu suatu kejadian (selesainya operasi I/O), maka proses menjadi *blocked* dan pemroses dialihkan ke proses lain.
3. Jika kwanta (*quantum*) belum habis tapi proses telah selesai, maka proses diakhiri dan pemroses dialihkan ke proses lain.

Algoritma ini sepenuhnya bergantung besarnya *time quantum*. Jika terlalu besar, algoritma ini akan sama saja dengan algoritma *first come first served*. Jika terlalu kecil, akan semakin banyak peralihan proses sehingga banyak waktu terbuang. Algoritma penjadwalan ini dapat diimplementasi sebagai berikut: – Mengelola senarai proses *read (runnable)* sesuai urutan kedatangan. – Ambil proses yang berada di ujung depan antrian menjadi *running*. – Bila quantum belum habis dan proses selesai maka

ambil proses di ujung depan antrian proses *ready*. – Jika quantum habis dan proses belum selesai maka tempatkan proses *running* ke ekor antrian proses *ready* dan ambil proses di ujung depan antrian proses *ready*.

2.3.2. Bentuk Algoritma

Algoritma ini menggilir proses yang ada di antrian. Proses akan mendapat jatah sebesar *time quantum*. Jika *time quantum*-nya habis atau proses sudah selesai, CPU akan dialokasikan ke proses berikutnya. Tentu proses ini cukup adil karena tak ada proses yang diprioritaskan, semua proses mendapat jatah waktu yang sama dari CPU yaitu $(1/n)$, dan tak akan menunggu lebih lama dari $(n-1)q$ dengan q adalah lama 1 quantum.

Algoritma ini sepenuhnya bergantung besarnya *time quantum*. Jika terlalu besar, algoritma ini akan sama saja dengan algoritma *first come first served*. Jika terlalu kecil, akan semakin banyak peralihan proses sehingga banyak waktu terbuang.

Permasalahan utama pada *Round Robin* adalah menentukan besarnya *time quantum*. Jika *time quantum* yang ditentukan terlalu kecil, maka sebagian besar proses tidak akan selesai dalam 1 quantum. Hal ini tidak baik karena akan terjadi banyak *switch*, padahal CPU memerlukan waktu untuk beralih dari suatu proses ke proses lain (disebut dengan *context switches time*). Sebaliknya, jika *time quantum* terlalu besar, algoritma *Round Robin* akan berjalan seperti algoritma *first come first served*. *Time quantum* yang ideal adalah jika 80% dari total proses memiliki CPU burst *time* yang lebih kecil dari 1 *time quantum* [6].

Berikut adalah algoritma penjadwalan *Round Robin* secara Keseluruhan :

Setiap proses mendapat jatah waktu CPU (*time slice/quantum*) tertentu *Time slice/quantum* umumnya antara 10 – 100 milidetik.

1. Setelah *time slice/quantum* maka proses akan di-*preempt* dan dipindahkan ke antrian *ready*.

2. Proses ini adil dan sangat sederhana.

Jika terdapat n proses di “antrian ready ” dan waktu quantum q (milidetik), maka:

1. Maka setiap proses akan mendapatkan $1/n$ dari waktu CPU.
2. Proses tidak akan menunggu lebih lama dari: $(n-1)q$ time units.

Performance dari algoritma ini tergantung dari ukuran time quantum:

1. Time Quantum dengan ukuran yang besar maka akan sama dengan FCFS.
2. Time Quantum dengan ukuran yang kecil maka time quantum harus diubah ukurannya lebih besar dengan respek pada context switch sebaliknya akan memerlukan ongkos yang besar.

III. METODE PENELITIAN

3.1. Teknik Penelitian

Proses standar yang harus digunakan dalam membangun aplikasi penjadwalan adalah:

1. Pengumpulan kebutuhan

Mendefinisikan format dan kebutuhan keseluruhan perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun prototyping

Membangun prototyping dengan membuat perancangan sementara yang berpusat pada penyajian (misalnya dengan membuat input dan contoh outputnya).

3. Evaluasi prototyping

Evaluasi ini dilakukan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka prototyping diperbaiki dengan mengulang langkah 1, 2, 3.

4. Mengkodekan system

Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji system

Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.

6. Evaluasi Sistem

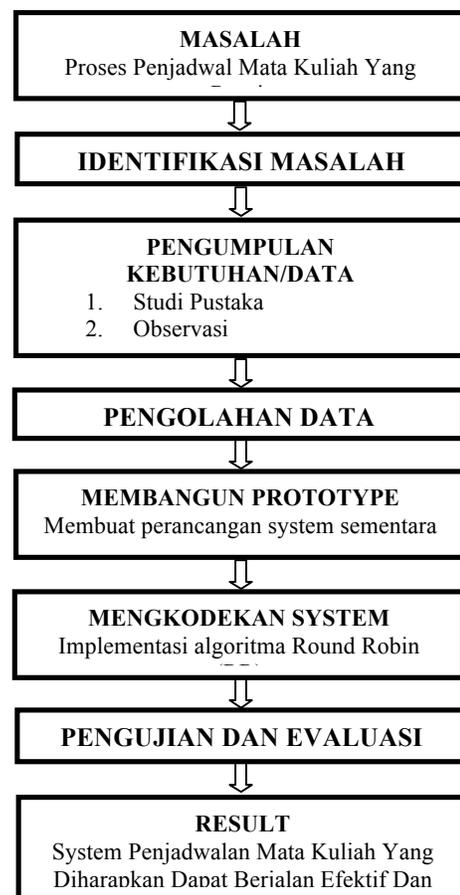
Mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.

7. Menggunakan system

Perangkat lunak yang telah diuji siap untuk digunakan.

3.2. Rancangan Penelitian

Gambar di bawah ini menjelaskan rancangan/kerangka kerja untuk penentuan penjadwalan mata kuliah.

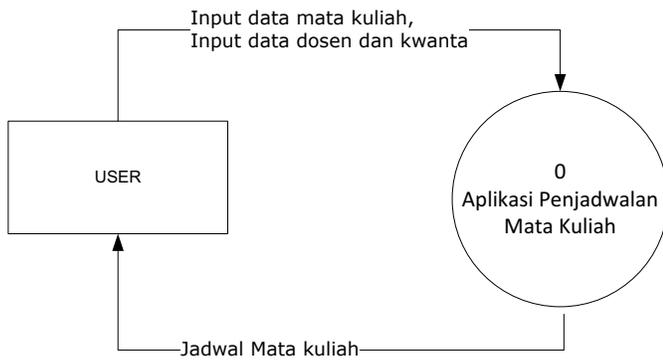


Gambar 3.1. Kerangka Kerja

3.3. Diagram Konteks (Context Diagram)

Untuk menggambarkan hubungan sistem dengan lingkungan luarnya diperlukan diagram konteks. Diagram

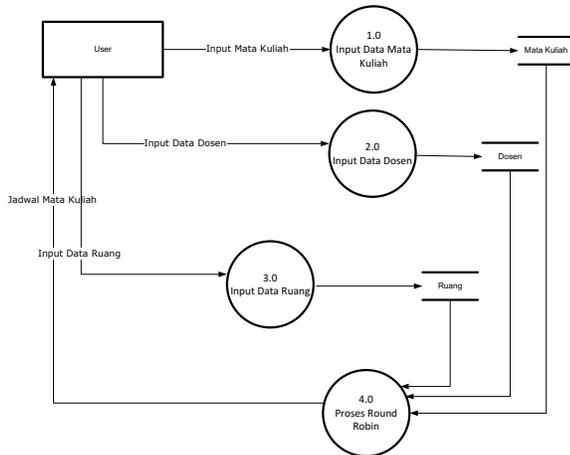
konteks dari sistem yang akan dirancang adalah seperti Gambar 3.2 dibawah ini



Gambar 3.2. Diagram Konteks

3.4. DFD Level 1 Proses 0 dan 1

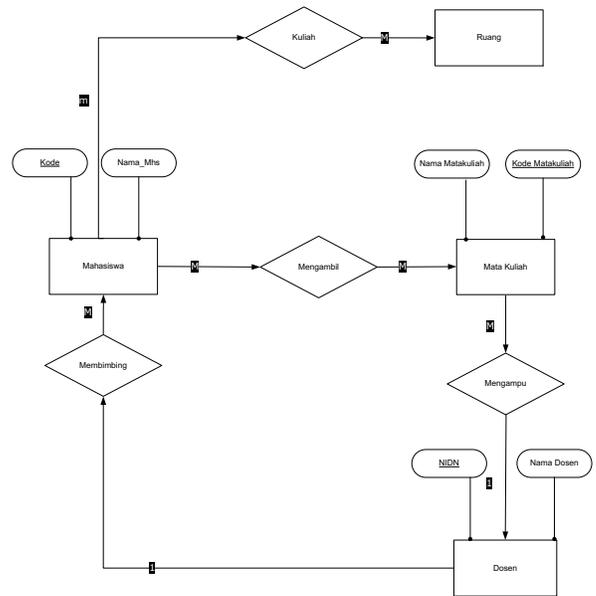
Diagram konteks pada Gambar 3.3 dapat diperinci menjadi DFD level-1. Proses-proses pada DFD level-1 merupakan dekomposisi dari proses pada diagram konteks. Proses-proses tersebut dapat dilihat pada Gambar 3.3.



Gambar 3.3. DFD Level 1 Proses 0 dan 1

3.5. ERD (Entity Relationship Diagram)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

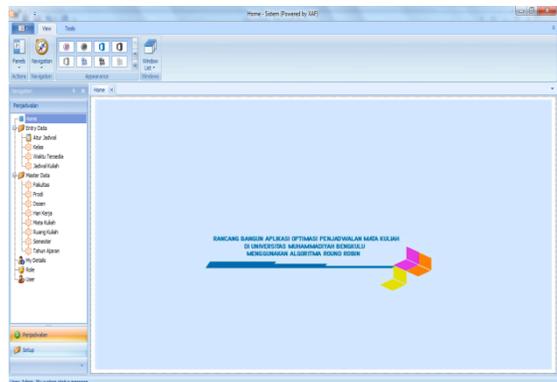


Gambar 3.4. ERD

IV. HASIL DAN PEMBAHASAN

4.1. Halaman Utama (Pembuka)

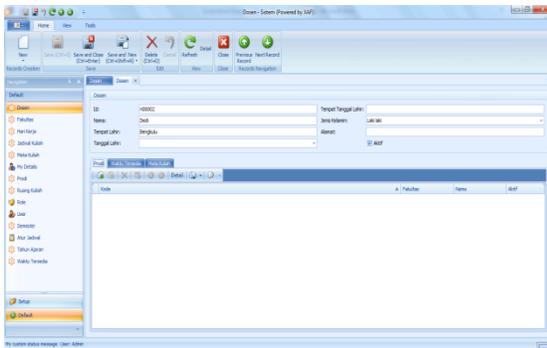
Halaman ini akan muncul pada pertama kali program di jalankan.



Gambar 4.1. Halaman Utama

4.2. Halaman Dosen

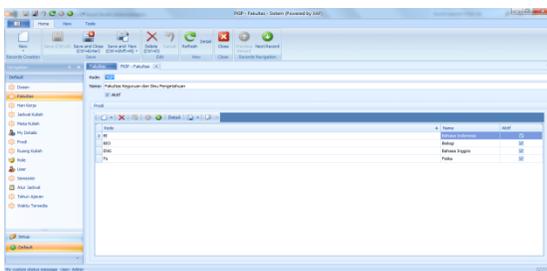
Halaman dosen, ini tersedia tambah edit dan hapus tentang semua data yang berhubungan dengan data dosen. Berikut tampilan halaman dosen.



Gambar 4.2. Halaman Menu Dosen

4.3. Halaman Fakultas

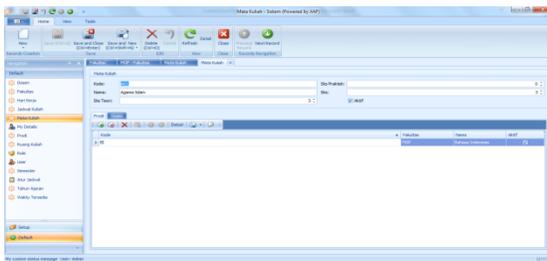
Pada Halaman ini kita dapat melakukan proses tambah, edit hapus data fakultas.



Gambar 4.3. Halaman Menu fakultas

4.4. Halaman Data Mata Kuliah

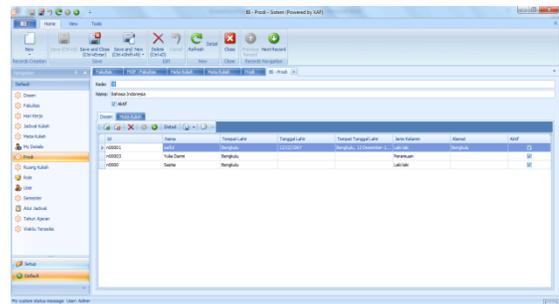
Pada Halaman ini kita dapat melakukan proses tambah, edit hapus data Mata kuliah.



Gambar 4.4. Menu Pengisian Mata Kuliah

4.5. Halaman Prodi

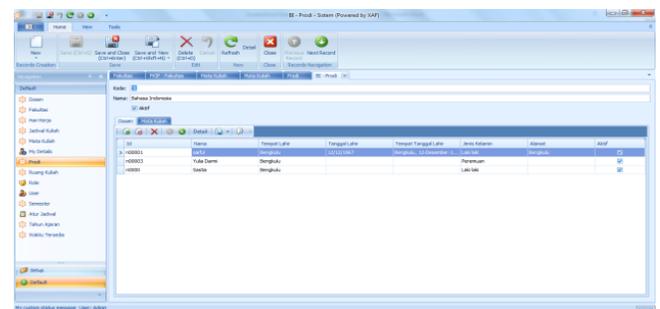
Pada Halaman ini kita dapat melakukan proses tambah, edit hapus data prodi.



Gambar 4.5. Menu Entri Daftar Prodi

4.6. Halaman Ruang kuliah

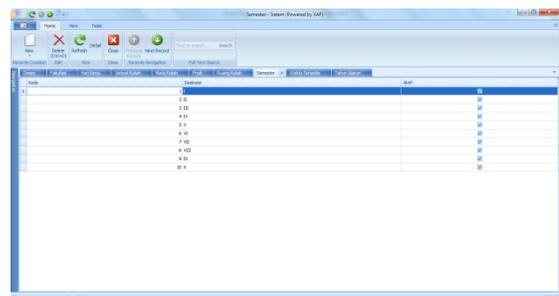
Pada Halaman ini kita dapat melakukan proses tambah, edit hapus data Ruang Kuliah. Sehingga proses pemilihan ruang kuliah akan dirandom oleh system. Dengan begini meminimalisir terjadinya tabrakan dalam penggunaan ruang kuliah.



Gambar 4.6. Halaman Entri Daftar Ruang Kuliah

4.7. Halaman Semester

Halaman ini akan muncul setelah kita tekan tombol semester pada aplikasi, halaman ini berfungsi mengatur mata kuliah disetiap semesternya.



Gambar 4.7. Halaman Menentukan Semester

4.8. Halaman Atur Jadwal

Halaman ini akan muncul setelah kita tekan tombol atur jadwal pada aplikasi, halaman ini berfungsi mengatur jadwal disetiap semesternya. Pada tahapan ini algoritma round robin akan bekerja dalam menyusun penjadwalan. Pada halaman ini harus diinputkan time dan waktu kuantu, selanjutnya klik tombol atur, jadwal akan diatur menggunakan algoritma round robin, setelah di klik maka akan muncul tampilan hasil pengaturan jadwal .



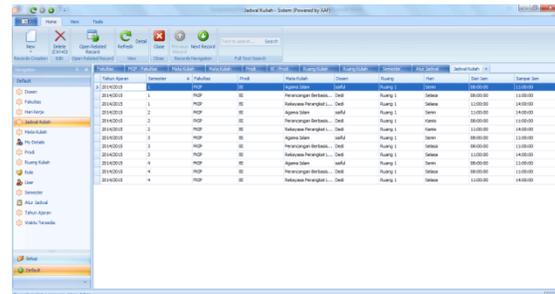
Gambar 4.8 Proses Atur Jadwal

Dalam algoritma penjadwalan proses Round Robin, proses akan diberikan porsi waktu pengerjaan yang sama dari tiap prosesnya. Algoritma Round Robin ini disebut juga dengan algoritma yang adil.

Pertama-tama data yang diproses untuk penjadwalan data pertama adalah data 1

1. Jika proses penjadwalan data 1 belum selesai dilakukan dan waktu kuantu belum habis, maka proses penjadwalan data 1 tetap berjalan.
2. Jika proses penjadwalan belum selesai dilakukan dan waktu kuantu sudah habis, maka proses penjadwalan dilanjutkan pada data 2 atau data selanjutnya, berdasarkan urutan dari data 1 sampai data 10
3. Jika proses penjadwalan sudah selesai dilakukan dan waktu kuantu belum habis, maka proses pengiriman dilanjutkan pada data 2 atau data selanjutnya, berdasarkan urutan dari data 1 sampai data 10.
4. Waktu kuantu digunakan untuk mengatur pengalihan proses dari satu data ke data lainnya.
5. Kecepatan digunakan untuk mempercepat atau memperlambat proses penjadwalan data, semakin besar nilai kecepatan maka pemrosesan data akan

semakin cepat dan semakin kecil nilai kecepatan maka pemrosesan data akan semakin lambat.



Gambar 4.9. Hasil Prose Atur Jadwal

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Dari hasil pembahasan, perancangan serta pembuatan aplikasi, maka penulis menarik kesimpulan sebagai berikut:

1. Penelitian ini berhasil menghasilkan sebuah aplikasi optimasi penjadwalan mata kuliah, aplikasi ini telah berhasil dijalankan dan memberikan *output* berupa hasil penjadwalan mata kuliah menggunakan algoritma round robin.
2. Dengan Adanya aplikasi ini maka pengguna dapat mengetahui jadwal mata kuliah setiap harinya pada setiap kelas dalam satu jurusan.

5.2. Saran

1. Dalam aplikasi ini masih ada diperlukan beberapa penyempurnaan. Sehingga diharapkan dapat kembali dikembangkan luasan.
2. Dalam aplikasi ini sifat yang digunakan masih bersifat offline sehingga bisa dikembangkan untuk menjadi online.

DAFTAR PUSTAKA

- [1] Puspaningrum, W. A., Djunaidy, A., & Fitriyani, R. A. (2013). Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika di Jurusan Sistem Informasi ITS. *JURNAL TEKNIK POMITS Vol. 2, No. 1, ISSN: 2337-3539 (2301-9271 Print)*, 5.
- [2] Santika, M., & Hansun, S. (2014). Implementasi Algoritma Shortest Job First dan Round Robin pada Sistem Penjadwalan Pengiriman Barang. *ULTIMATICS, Vol. VI, No. 2*, 6.
- [3] Rifqo, M. H., & Wijaya, A. (2016). Implementasi Algoritma Naive Bayes dalam Penentuan Pemberian Kredit. *PDP Ristek Dikti*, 15.
- [4] Munir, R. (2004). *Algoritma dan Pemrograman Dalam Bahasa Pascal dan C*. Bandung: Informatika Bandung.
- [5] Lumbantoruan, G. (2016). MODIFIKASI ALGORITMA ROUND ROBIN DENGAN DYNAMIC QUANTUM TIME DAN PENGURUTAN PROSES SECARA ASCENDING. *Jurnal ISD Vol.2 No.2 ISSN : 2528-5114*, 12.
- [6] Triono, G. (2015). Implementasi Load Balancing Dengan Menggunakan Algoritma Round Robin Pada Kasus Pendaftaran Siswa Baru Sekolah Menengah Pertama LABSCHOOL UNESA SURABAYA. *IDeaTech ISSN: 2089-1121*, 8.