

# Aplikasi Penjadwalan Sidang Tugas Akhir di Fakultas Teknologi Informasi Universitas Kristen Maranatha Dengan Metode Depth First Search

<http://dx.doi.org/10.28932/jutisi.v4i1.760>

Yulius Bastian<sup>#1</sup>, Daniel Jahja Surjawan<sup>\*2</sup>, Adelia<sup>#3</sup>

<sup>#1</sup>Program Studi Teknik Informatika, Universitas Kristen Maranatha  
Jl. Prof. Drg. Surya Sumantri no. 65, Sukawarna, Bandung

<sup>#3</sup>Program Studi Sistem Informasi, Universitas Kristen Maranatha  
Jl. Prof. Drg. Surya Sumantri no. 65, Sukawarna, Bandung

<sup>1</sup>yulius.bastian@hotmail.com

<sup>3</sup>adelia@it.maranatha.edu

<sup>\*2</sup>Program Studi Teknik Informatika, Universitas Kristen Maranatha  
Jl. Prof. Drg. Surya Sumantri no. 65, Sukawarna, Bandung

<sup>2</sup>daniel.js@it.maranatha.edu

**Abstract** — Every student in all Universities who is already in the final semester will report the results in the form of presentation in the presence of mentors and examiners. Students will usually wait for a schedule that has been determined by the coordinator in the Faculty. This scheduling sometimes makes students anxious because they have to wait until the time is appropriate and often they don't get a suitable schedule. This also happens with the students in Faculty of Information Technology, Maranatha Christian University. The coordinators often experience difficulties in the allocation of mentors or examiners to be scheduled for a council. This research will make an application that help in conducting scheduling of the council so that the council process can be held smoothly and students will not be afraid to miss the schedule. Applications will be built using JavaFX desktop programming, JSON type network services, and data reporting through JasperReport. Data were collected in literature and interviews with related parties. Applications will accommodate from data entry, role assignment determination, scheduling, and reporting. Applications use data exchange methods through network services that facilitate the development of data needs. This prototype has been tested by black box method. The test results state that this prototype works through testing with all of the possibilities.

**Keywords**— Mentor, Examiner Allocation, Reporting, Scheduling.

## I. PENDAHULUAN

Pada semua Universitas, setiap mahasiswa yang sudah pada tingkat akhir perkuliahan tentunya akan melaporkan hasilnya dalam bentuk presentasi di hadapan pembimbing dan penguji. Mahasiswa biasanya akan menunggu jadwal

yang sudah ditentukan oleh Fakultas. Penjadwalan ini terkadang membuat mahasiswa cemas karena mahasiswa harus menunggu sampai kesediaan waktu yang sesuai, dan bahkan seringkali tidak ditemukan jadwal yang cocok. Hal ini juga terjadi pada Fakultas Teknologi Informasi, Universitas Kristen Maranatha, koordinator seringkali mengalami kesulitan dalam hal alokasi pembimbing atau penguji yang akan dijadwalkan untuk sidang.

Proses yang terjadi saat ini dimulai dari pengumpulan data mahasiswa yang mengajukan topik kepada koordinator. Data diperoleh dari masing-masing Kelompok Bidang Keahlian (KBK) dimana terdapat 3 KBK yaitu Multimedia & Jaringan, Rekayasa Perangkat Lunak, dan Sistem Informasi. Masalah terjadi disebabkan karena masing-masing KBK tidak memiliki standar format yang sama dalam melakukan input data mahasiswa. Setiap KBK melaporkan masing-masing mahasiswa yang mengajukan topik atau sidang dengan format laporan yang berbeda. Pencatatan yang digunakan saat ini masih manual menggunakan aplikasi spreadsheet dalam pengelolaan datanya. Tidak adanya data tentang jumlah mahasiswa bimbingan yang sedang ditangani bisa mengakibatkan seorang dosen melebihi batas ketentuan.

Setiap sidang membutuhkan pasangan pembimbing dan penguji. Pasangan pembimbing dan penguji ini harus memiliki alokasi waktu senggang yang sama untuk melakukan sidang. Sistem penjadwalan yang digunakan sekarang masih manual yaitu dengan melihat dan bertanya langsung ke masing-masing KBK. Aplikasi yang akan dibangun diharapkan dapat membantu dalam penjadwalan sidang. Para dosen akan memasukkan alokasi waktu senggang ke dalam aplikasi. Melalui mekanisme yang ada

akan dicari kecocokan jadwal antara mahasiswa, pembimbing dan penguji. Jika jadwal dosen pembimbing dengan semua dosen penguji tidak terdapat kecocokan, maka dosen yang bersangkutan diminta untuk mengubah atau menambah kesediaan alokasi waktunya sampai ditemukan waktu yang sesuai.

Setelah ditemukan waktu yang cocok antara mahasiswa, pembimbing, dan penguji, selanjutnya setiap KBK harus membuat laporan hasil dari sidang yang telah berlangsung. Laporan dibagi menjadi beberapa bagian seperti mahasiswa yang lulus atau tidak lulus, perpanjangan atau tidak perpanjangan topik dan peserta setiap semester beserta statusnya. Laporan akan dibuat secara otomatis melalui sistem yang sudah menyiapkan kategori laporan yang dibutuhkan sehingga format laporan pun akan seragam antar KBK yang ada.

Dengan dibuatnya aplikasi ini diharapkan dapat membantu setiap koordinator dalam melakukan alokasi, penjadwalan dan pelaporan yang terjadi di Fakultas Teknologi Informasi, sehingga tenaga dan waktu yang ada dapat dialokasikan untuk hal lain guna meningkatkan efektivitas serta produktivitas instansi terkait.

## II. KAJIAN TEORI

Kajian ini menjadikan beberapa konsep dan teori sebagai landasan penelitian. Konsep dan teori yang digunakan antara lain *web service*, *reporting system*, dan JUnit Test.

### A. Sistem Informasi

Sistem informasi adalah keterhubungan elemen atau komponen yang mengambil masukan (*input*), manipulasi proses (*process*), menyimpan / menyebarkan / menghasilkan sesuatu (*output*) data atau informasi dan menyediakan tanggapan (*feedback*). *Input* adalah aktivitas di mana mengambil data mentah. Data tersebut kemudian di proses yang melibatkan konversi atau transformasi menjadi keluaran yang berguna. Keluaran / output biasanya dalam bentuk dokumen maupun laporan. Sementara umpan balik dibutuhkan untuk membuat perubahan pada masukan data atau alur proses yang ada. Umpan balik juga penting bagi manajer dan pembuat keputusan [1].

Sistem memiliki beberapa karakteristik atau sifat yang terdiri dari komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengolahan sistem dan sasaran sistem. Jadi, sistem informasi adalah hubungan antar unit yang saling berhubungan, yang digunakan untuk tujuan tertentu [2]

### B. Unified Modeling Language

UML adalah standar bahasa untuk membuat cetak biru sebuah perangkat lunak. UML digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak yang intensif. Arsitek perangkat lunak membuat diagram UML untuk membantu pengembang membangun perangkat lunak [3].

### C. Relational Database

Membangun suatu relational database sangat tergantung pada kemampuan dalam menetapkan suatu model relational. Model harus benar-benar menggambarkan penuh bagaimana data diorganisir dalam ketentuan struktur data, integriti, query, manipulasi, dan penyimpanan.

Relational database memungkinkan user-user untuk mendefinisikan kolom-kolom record tertentu sebagai key atau index, melakukan pencarian, menghubungkan record-record antar tabel dan menetapkan batasan-batasan integritas. Query-query pencarian lebih cepat dan lebih akurat jika berdasarkan nilai-nilai yang telah diindex. Record-record tabel dapat dengan mudah dihubungkan dengan nilai-nilai yang telah diindex. Batasan-batasan integritas dapat ditetapkan untuk menjamin bahwa hubungan antar tabel tersebut. Relational database menawarkan proses reporting yang lebih baik, dengan berbagai report generator yang memfilter dan menampilkan kolom-kolom pilihan. Relational database menawarkan kemampuan membuat module-module reporting. Sebagian besar relational database juga menawarkan kemampuan mengimpor dan mengeksport data dari software lainnya.

Terdapat tiga sistem relational database yang utama: proprietary, open source, dan embedded. Relational database yang proprietary biasanya memerlukan penggunaan bahasa-bahasa pengembangan yang juga bersifat proprietary guna menyempurnakan SQL. Sebagai contoh MS Access yang menggabungkan visual basic dengan SQL. Databasedatabase open source, seperti MySQL, didistribusikan dengan gratis agar mendorong pengembangan user. Embedded dan relational database dikemas sebagai bagian dari paket-paket software lainnya, seperti paket-paket software tax-preparation. Vendor mensuplai database, dan seluruh perlengkapan-perengkapan manipulasi yang berhubungan, untuk mengontrol struktur database. Database-database ini biasanya disertakan oleh perlengkapan-perengkapan yang mana menyediakan audit bekas-bekas transaksi [4].

### D. Web Service

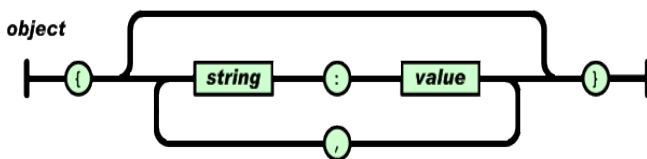
Penaafsiran tentang *web service* beragam dan mempunyai makna yang terus berkembang. Namun dapat di lihat dari ragamnya bahwa *web service* adalah jenis dari *webified application* di mana komunikasinya melalui HTTP. Web Service secara kasar dapat dibagi menjadi dua kelompok, yaitu SOAP-based dan REST-style. SOAP sekarang sudah menggeser perannya menjadi Service Oriented Architecture Protocol. SOAP hanya berupa XML yang di dalamnya terkandung pesan. SOAP mempunyai standard, toolkit dan libraries sedangkan REST terlihat seperti bentuk sederhana dari kompleksitas web service SOAP-based. Baik SOAP maupun REST keduanya hanya akan tampak seperti program tanpa desain grafis antarmuka [5].

*Client* dari sebuah *web service* dapat dibuat dari berbagai macam bahasa pemrograman yang memiliki *library*

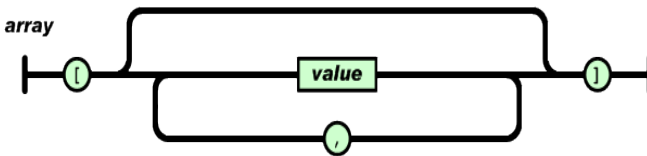
pendukung, jadi tidak perlu disamakan penggunaan bahasa pemrograman pada *web service* dan *client* [5]. Pola kerja pada *SOAP-based* dengan *REST-style web service* terdapat perbedaan. *SOAP-based* akan mengirimkan SOAP document sebagai request pada *web service*, kemudian dikembalikan pula dalam bentuk *SOAP document* sebagai balasan. Sementara pada layanan *REST-style*, *client* hanya mengirimkan standard http request kemudian dikembalikan dalam bentuk XML sebagai balasan.

JSON adalah salah satu format dari web service kelompok REST. JSON adalah format pertukaran data yang sangat sering digunakan untuk komunikasi antara aplikasi dengan perangkat melalui internet. Format pertukaran data JSON lebih ringan dan lebih cepat untuk di proses dibandingkan dengan format XML. Bahasa pemrograman seperti C, C++, C#, Java, JavaScript, Perl, Python dan lainnya dapat menggunakan format JSON ini karena sifatnya yang tidak mengikat pada bahasa pemrograman tertentu [6].

Bentuk penulisan JSON terdiri atas kunci dan nilai [7]. Pada Pemrograman Java isinya seperti HashMap. Perbedaannya hanya pada tipe data yang digunakan dalam JSON adalah String, yang kemudian diubah menjadi objek pada setiap pasang kunci dan nilainya. Sebuah kunci juga dapat memiliki nilai berbentuk larik. Bisa juga dalam sebuah kunci, terdapat pasangan kunci dan nilai. Kembalian data dari *web service* dengan format JSON banyak menggunakan *delimiter* dan *escape characters*. Hal ini diperlukan untuk memisahkan antara kunci dengan nilai. Beberapa bentuk JSON secara visual di mana representasi bentuknya sebagai objek, larik dan nilai dapat dilihat pada Gambar 1 dan Gambar 2 berikut.



Gambar 1 JSON Bentuk objek [7]



Gambar 2 JSON Bentuk larik [7]

### E. JavaFX

*JavaFX* merupakan *framework* pengganti *Swing* untuk mengembangkan program Java yang menggunakan GUI [8]. *JavaFX* dalam pengembangannya secara *default* menerapkan pola MVC. Pola ini bertujuan untuk memisahkan antara data (*model*), tampilan (*view*) dengan kendali atas tampilan (*controller*) [9]. MVC mengikuti pendekatan yang paling umum dari pemisahan logika membagi kode ke dalam bentuk kelas-kelas berbeda [10].

Sedangkan pada *framework* *Swing* secara *default* tidak dipisahkan antara *model*, *view* dan *controller*.

*Stage* adalah sebuah komponen dalam *JavaFX* yang akan menampilkan antarmuka pengguna [11]. Sementara *Scene* adalah komponen tempat menampung komponen grafis yang akan ditampilkan pada *stage* [11]. Tampilan antarmuka dapat dibuat melalui dua cara, yaitu dengan menggunakan kode Java atau menggunakan kode *FXML*. Pembuatan tampilan dengan kode *FXML* dapat menggunakan perangkat *JavaFX Scene Builder* versi 2.0.

Penggunaan perangkat *JavaFX Scene Builder* sama seperti penggunaan pada *Swing*, yaitu *drag and drop*. *Scene Builder* memiliki perhitungan yang lebih baik untuk peletakan komponen dibandingkan dengan *Swing*. Hal ini berdampak pada proporsional tampilan yang lebih baik. Penamaan komponen dalam *Scene Builder* ini disebut dengan *fxId*. Sehingga untuk mengakses properti komponen *JavaFX* perlu memberikan *fxId* pada setiap komponen yang dibutuhkan.

### F. JUnit Test

Kode sumber yang tidak diuji adalah kode sumber yang buruk. Tidak peduli kode itu ditulis dengan baik, menggunakan orientasi objek dan sebagainya. Tanpa pengujian kita tidak tahu kualitas kode sumber yang dibuat semakin baik atau buruk.

Kerangka kerja pengujian yang lumrah pada pemrograman Java salah satunya *JUnit*. Merupakan perangkat pengujian yang berlisensi open-source dan hingga saat ini menjadi perangkat standar pengujian kode Java. Perangkat pengujian ini didukung oleh hampir semua IDE seperti *Netbeans*, *Eclipse* dan *IntelliJ IDEA*.

Konsep dari unit test adalah untuk memastikan class yang digunakan berjalan dengan semestinya. Prosesnya meliputi memasukkan data yang akan ditanggapi dengan hasil yang diharapkan (*expected output*), masukkan data yang tidak beraturan maka akan mengeluarkan eksepsi yang sesuai dan sebagainya [12].

### G. Hypertext Preprocessor

*PHP* (*Hypertext Preprocessor*) merupakan sebuah open source server scripting language yang tentunya akan dieksekusi dalam sebuah server. Sebuah berkas *PHP* dapat berisi teks, *HTML*, *CSS*, *JavaScript*, dan *PHP* itu sendiri. *PHP* dapat membuat isi halaman menjadi dinamis, membaca hingga mengubah berkas yang ada pada server, mengambil data, memodifikasi data dalam database, dan mengenkripsi data. Berkas bentuk *PHP* dapat diakses oleh hampir semua platform dan kompatibel dengan hampir semua server [13]

## III. ANALISIS DAN RANCANGAN SISTEM

Perancangan dimulai dari pemetaan proses bisnis yang berlangsung sehingga didapatkan model pengembangan untuk pencarian solusi.

A. Proses Bisnis

1. Pendaftaran Topik dan Pengajuan PIC

Mahasiswa yang sedang mengambil mata kuliah STA/TA akan membuat draf proposal yang berisikan topik dan deskripsi topik secara global. Draft yang telah dibuat diserahkan kepada koordinator STA/TA. Koordinator STA/TA akan berkoordinasi dengan ketua KBK yang terdiri dari KBK SI, RPL, dan MM-Jar. Koordinasi antara ketua KBK akan menghasilkan pengelompokan topik yang telah disusun oleh mahasiswa ke dalam KBK yang sesuai. Setelah mencapai kesepakatan maka koordinator akan memasukkan data pasangan topik serta KBK ke dalam sistem. Setelah itu ketua KBK akan mendapatkan daftar yang sesuai dengan KBK yang kemudian digunakan untuk menentukan PIC mahasiswa bersangkutan seperti pada Gambar 3.

2. Perizinan Maju Sidang

Peserta yang sudah siap untuk maju sidang akan membuat pernyataan yang disetujui oleh pembimbing. Pernyataan tersebut diserahkan kepada koordinator kemudian akan dibukakan akses untuk penjadwalan sidang

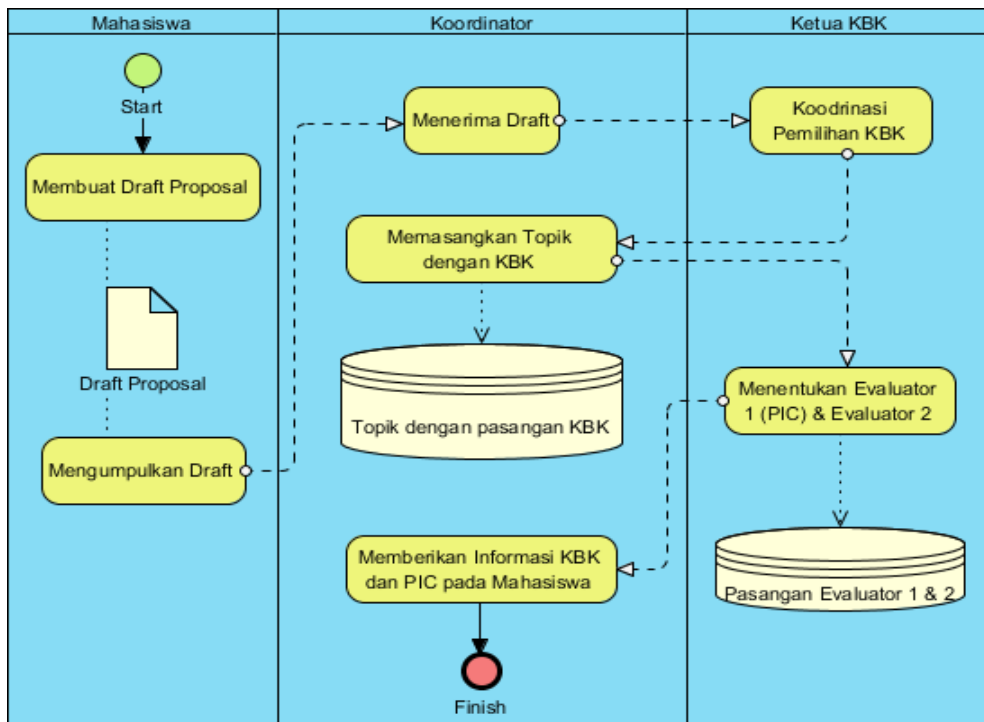
oleh ketua KBK. Jika peserta tersebut mengajukan pra-sidang maupun USTA, maka ketua KBK akan terlebih dahulu menentukan penguji tambahan untuk sidang tersebut seperti pada **Kesalahan! Sumber referensi tidak ditemukan..**

3. Pengajuan Jadwal Berhalangan Sidang

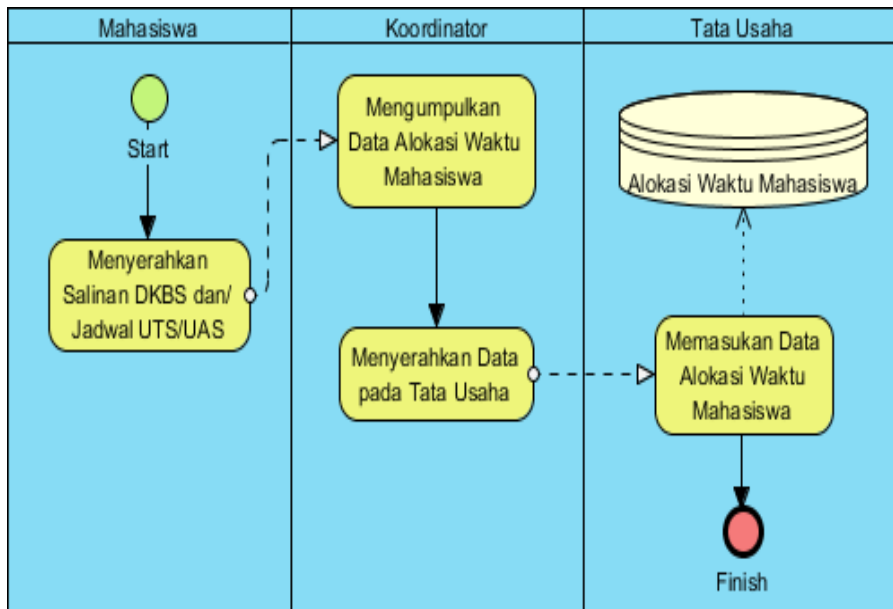
Semua dosen pada kondisi sebelum masa pengajuan izin sidang harus memasukkan terlebih dahulu alokasi waktu yang bisa dipakai untuk persidangan. Mahasiswa pun harus memberikan jadwal kuliah serta jadwal berhalangan lainnya kepada petugas tata usaha melalui koordinator untuk di masukan ke dalam sistem seperti pada Gambar 4.

4. Penyelesaian Sidang

Setelah masa persidangan selesai koordinator akan mendapat status kelulusan mahasiswa yang menjalani sidang. Status tersebut kemudian di masukan ke dalam sistem kemudian Ketua KBK dapat membuat laporan hasil sidang dalam periode tersebut per KBK yang ada seperti pada **Kesalahan! Sumber referensi tidak ditemukan..**



Gambar 3 Alur pendaftaran topik dan penunjukan PIC



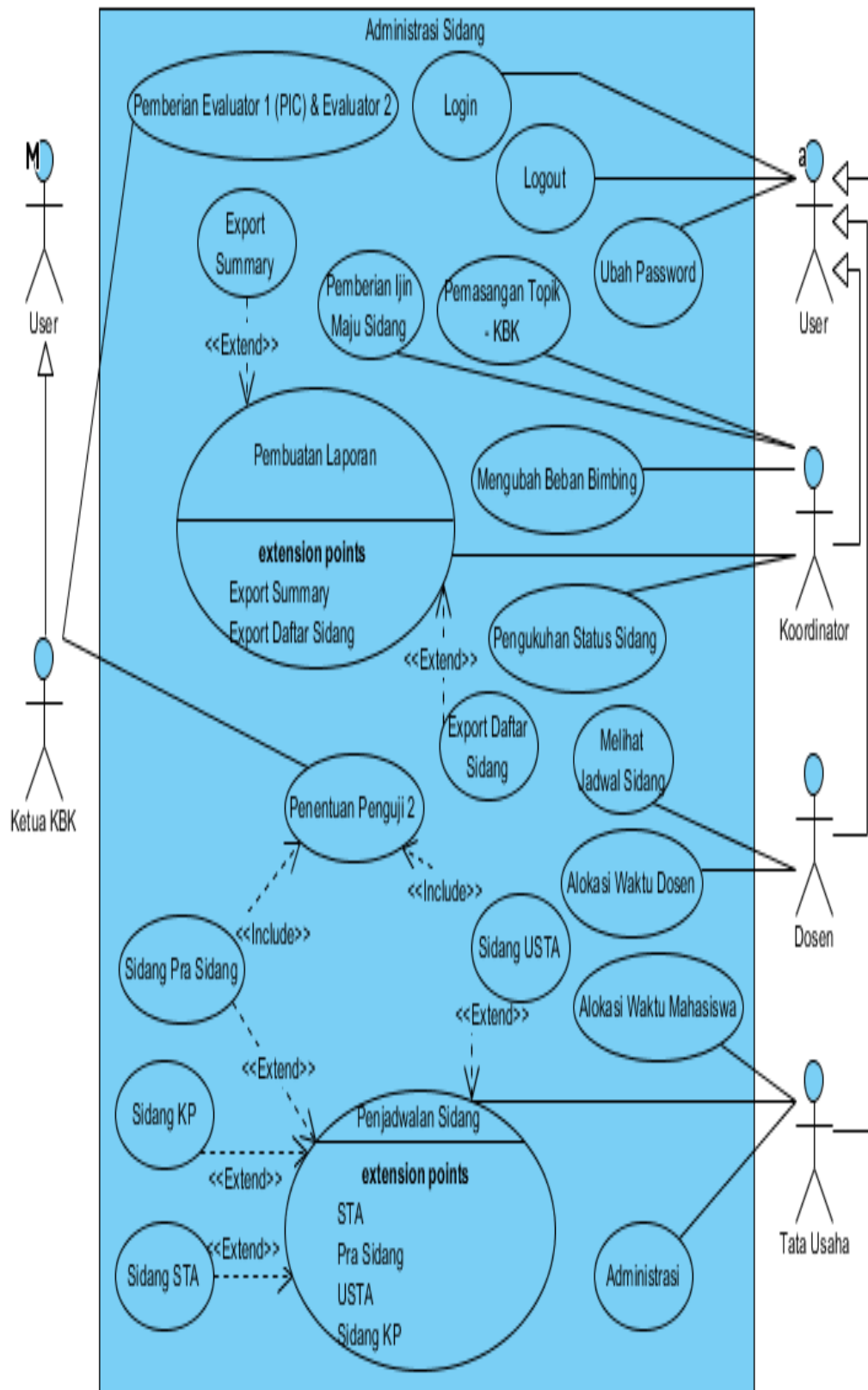
Gambar 4 Alur pengajuan jadwal berhalangan sidang

**B. Rancangan Aplikasi**

Pengguna aplikasi ini terdiri dari empat jenis yaitu Dosen, Ketua KBK, Koordinator, dan Tata Usaha. Setiap pengguna mempunyai hak administrasi masing-masing yang terlihat distribusinya dalam use case pada Gambar 5.

**C. Algoritma Penjadwalan**

Penjadwalan sidang pada aplikasi ini menggunakan dasar algoritma *Depth-First Search* (DFS). Data yang ada perlu disesuaikan terlebih dahulu dengan menggunakan prinsip clustering agar dapat diproses oleh algoritma DFS. Setelah data tersebut berubah menjadi model tree maka pencarian menggunakan DFS dapat dilakukan.



Gambar 5 Use Case

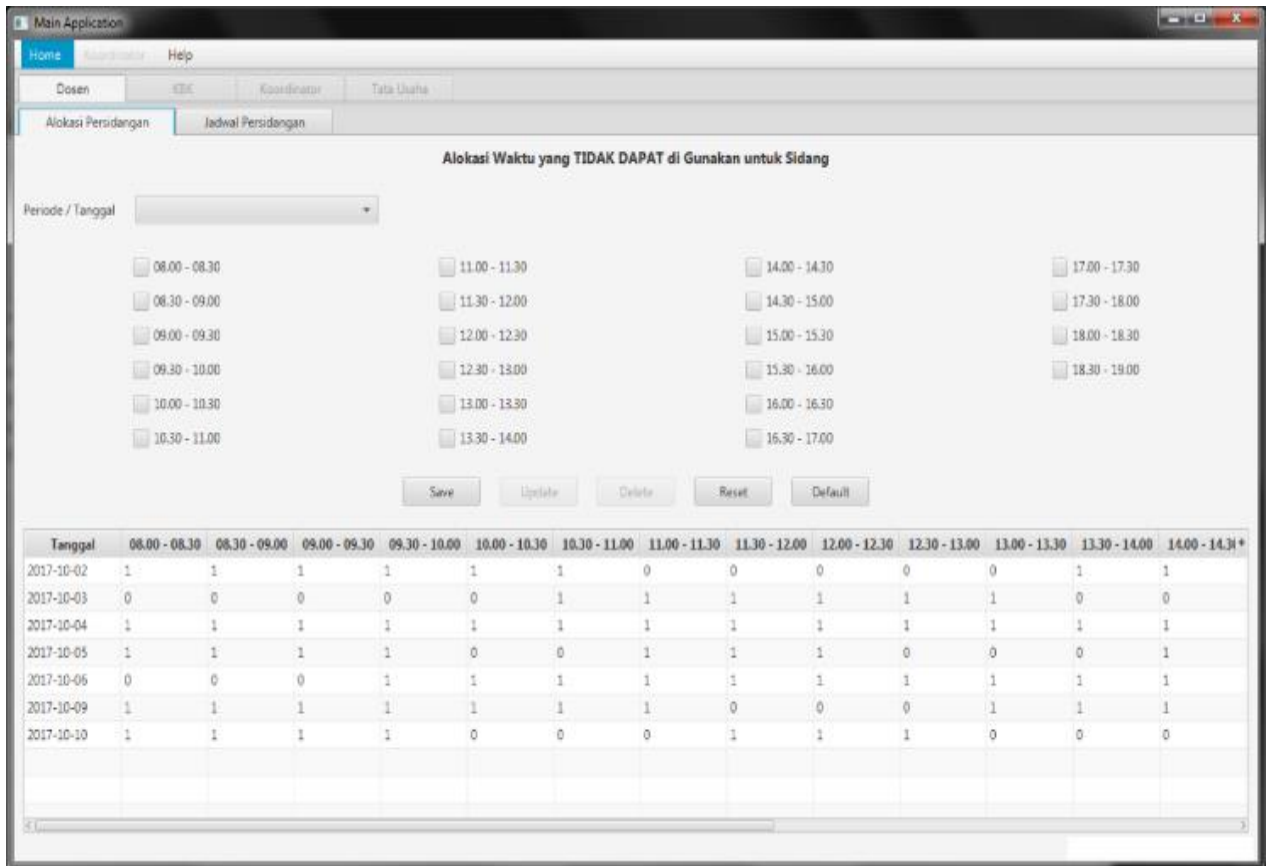
IV. IMPLEMENTASI

A. Pengguna sebagai Dosen

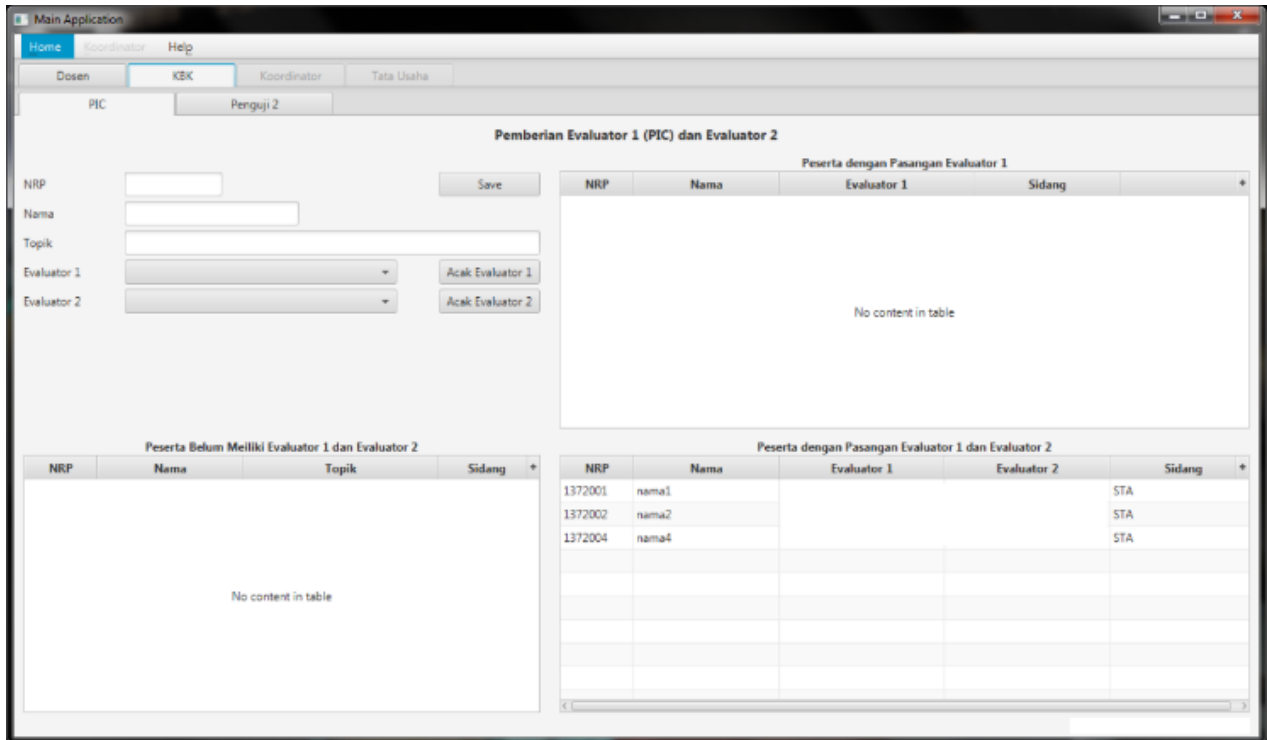
Dosen yang tidak mempunyai jabatan / peran khusus dalam persidangan STA/TA hanya bisa mengakses jendela Dosen, sementara jendela lainnya tidak. Pada jendela Dosen seorang dosen dapat mengatur alokasi waktu yang akan digunakan untuk persidangan seperti pada Gambar 6 dan melihat jadwal persidangan yang telah dibuat oleh tata usaha seperti pada **Kesalahan! Sumber referensi tidak ditemukan..**

B. Pengguna sebagai Ketua KBK

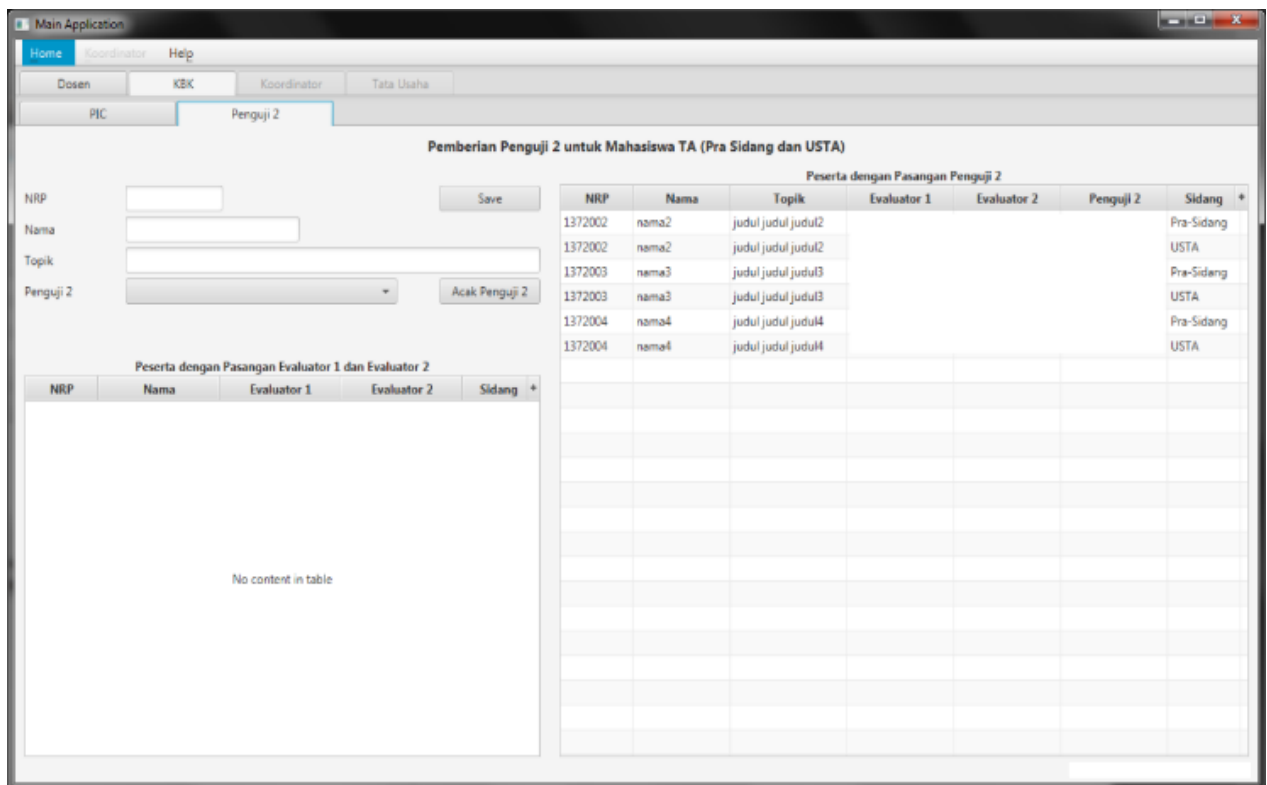
Jendela pengguna sebagai KBK memiliki beberapa sub jendela yang memiliki fungsi berbeda-beda. Terdapat jendela untuk penentuan PIC seperti pada Gambar 7 dan pengujian tambahan seperti pada Gambar 8. Tampilan awal secara default adalah pada jendela penentuan PIC. Pengguna sebagai KBK dapat mengakses jendela Dosen untuk memasukkan data ketersediaan melakukan sidang.



Gambar 6 Alokasi persidangan Dosen



Gambar 7 Penentuan PIC



Gambar 8 Penentuan penguji tambahan (Penguji 2)

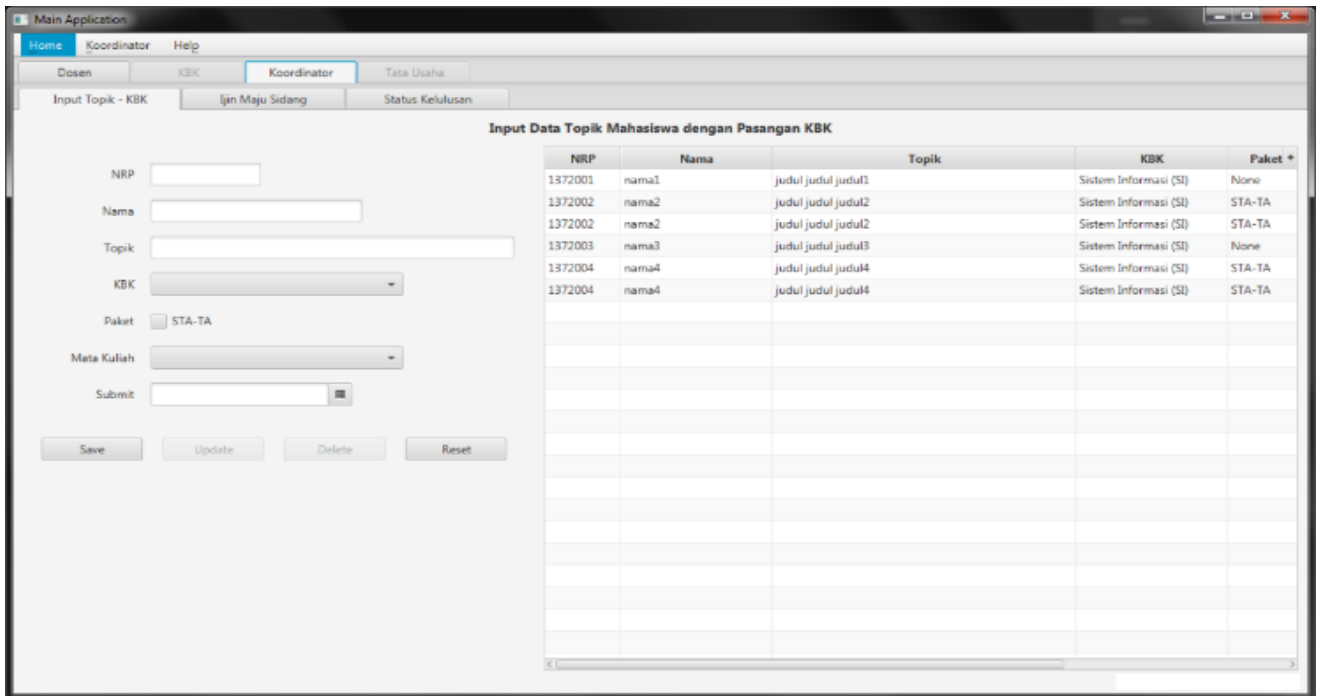


C. Pengguna sebagai Koordinator

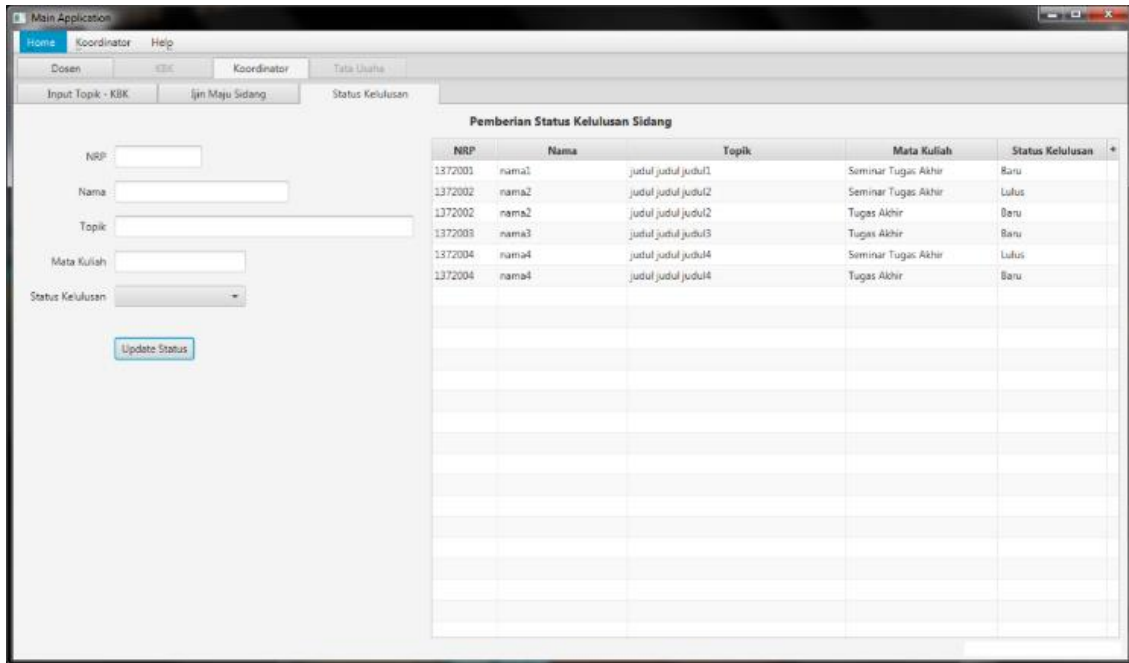
Jendela pengguna sebagai Koordinator memiliki beberapa sub jendela yang memiliki fungsi berbeda-beda. Terdapat jendela untuk memasukkan pasangan topik-KBK yang bersangkutan seperti pada Gambar 9, pemberian ijin untuk maju sidang seperti pada **Kesalahan! Sumber referensi tidak ditemukan.**, penetapan status dan kelulusan sidang seperti pada Gambar 10.. Tampilan awal secara default adalah pada jendela pemasangan topik-KBK. Pengguna sebagai Koordinator dapat mengakses jendela Dosen untuk memasukkan data ketersediaan melakukan sidang.

D. Pengguna sebagai Tata Usaha

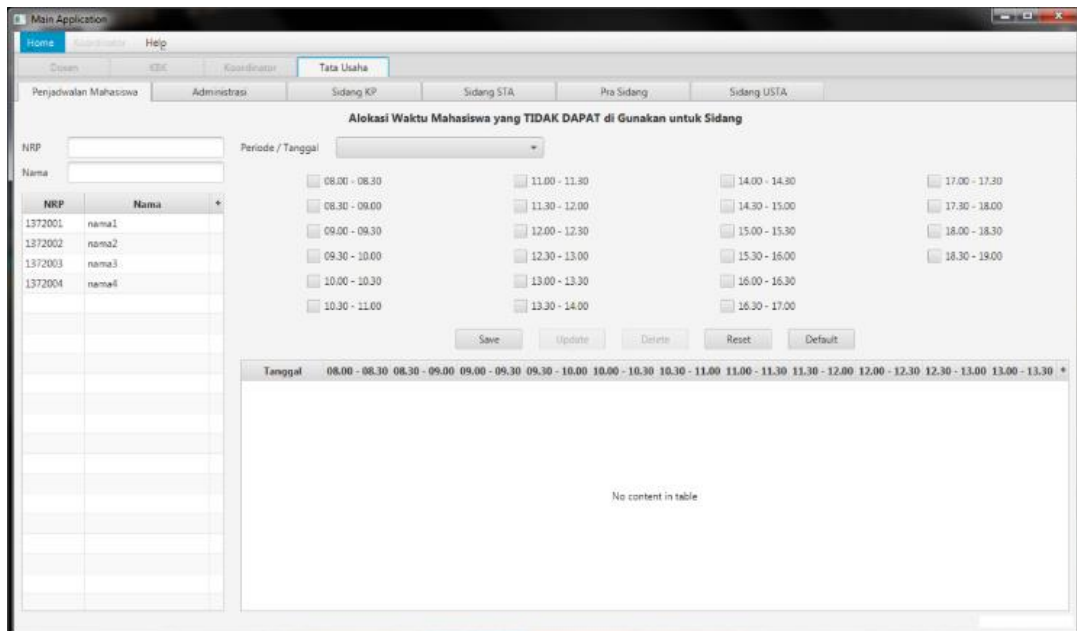
Pengguna yang masuk sebagai tata usaha bertugas mengelola tahun akademik dan periode sidang seperti pada **Kesalahan! Sumber referensi tidak ditemukan.**, pengalokasian waktu mahasiswa seperti pada Gambar 11 dan penjadwalan sidang mahasiswa seperti pada Gambar 12. Awal dari proses persidangan berada pada tata usaha di mana harus mempersiapkan terlebih dahulu tahun akademik kemudian dilanjutkan dengan masa persidangan yang dilakukan pada jendela Administrasi.



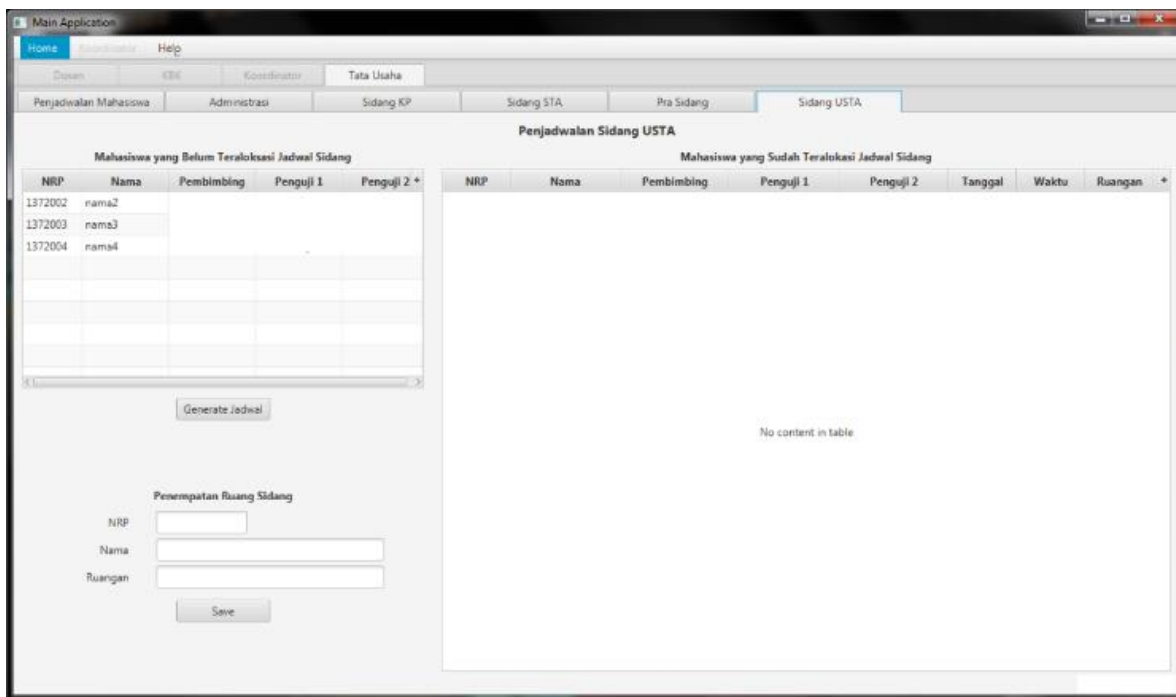
Gambar 9 Input Topik dengan pasangan KBK



Gambar 10 Penetapan status kelulusan persidangan mahasiswa



Gambar 11 Alokasi waktu persidangan mahasiswa



Gambar 12 Penjadwalan sidang USTA

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Kesimpulan memuat manfaat dari aplikasi yang sudah dibangun dan ditemukan selama proses pengembangan. Simpulan tersebut di susun dalam butir-butir sebagai berikut.

1. Aplikasi menghasilkan format masukkan data yang seragam bagi para koordinator. Selama menggunakan aplikasi yang sama integritas data akan terjamin walaupun penggunanya banyak.
2. Kontrol terhadap batasan maksimal beban pembimbing sekarang lebih mudah diatur oleh aplikasi. Pengguna yang bertugas mendistribusikan beban bimbingan tidak perlu lagi menghitung secara manual.
3. Pada dasarnya proses mencari kecocokan jadwal secara manual sama prosesnya dengan sistem yang terotomatisasi. Satu per satu komponen di periksa ketersediaannya kemudian di pasang. Proses pencocokan jadwal dengan aplikasi ini waktu eksekusinya lebih cepat dibandingkan dengan proses yang dilakukan secara manual.
4. Laporan yang dibutuhkan dari proses ini dapat dibuat secara langsung menggunakan perangkat pembuat laporan yang dapat mengekstrak data dari pangkalan data menjadi sebuah laporan. Pengambilan data dengan mengakses pangkalan data membuat pengguna dapat meminta data apa pun yang ada dalam pangkalan data untuk dijadikan sebuah laporan atau pun bentuk keluaran lainnya.

### B. Saran

Berikut ini adalah saran yang dapat digunakan untuk pengembangan kedepannya.

1. Masukkan data dan administrasi dapat dilakukan menggunakan perangkat smartphone (web browser / mobile apps) sehingga tidak terpaku pada computer di ruang dosen.
2. Fitur percakapan dalam aplikasi dapat dipertimbangkan sehingga pengelola tidak perlu melakukan perbincangan mengenai administrasi persidangan dengan menggunakan aplikasi lainnya. Fitur pengiriman jadwal melalui email kepada masing-masing dosen juga dapat ditambahkan sehingga dosen tidak perlu mengakses aplikasi untuk melihat jadwal persidangan yang menyangkut dirinya.
3. Data mahasiswa yang ada dalam sistem harus terintegritas dengan sistem terpadu. Pada kasus ini UK Maranatha mempunyai SAT sebagai pangkalan data utama. SAT dapat digunakan untuk menjaga integritas data dalam sistem.
4. Penentuan ruangan yang masih manual dapat dijadikan otomatis bila data ketersediaan ruangan telah disediakan. Menggunakan algoritma pencarian tertentu untuk mencari ketersediaan ruangan yang dikombinasikan dengan jadwal yang telah dibuat.

DAFTAR PUSTAKA

- [1] R. M. Stair and G. W. Reynolds, Principles of Information Systems, Course Technology, 2010.
- [2] Sutarman, Buku Pengantar Teknologi Informasi, Jakarta: Bumi Aksara, 2012.
- [3] R. S. Pressman, Software Engineering : A Practitioner's Approach — 7th ed., New York: McGraw-Hill, 2010.
- [4] MySQL, "MySQL 5.7 Reference Manual," [Online]. Available: <https://dev.mysql.com/doc/refman/5.7/en/introduction.html>. [Accessed September 2017].
- [5] M. Kalin, Java Web Service: Up and Running, Sebastopol: O'Reilly Media, Inc., 2009.
- [6] H. S. Padda and D. G. K. Gupta, "Analysing Impact of Delimiters on the Size of JSON Data Interchange Format," *International Research Journal of Engineering and Technology (IRJET)*, vol. 2, no. 8, pp. 453-456, 2015.
- [7] "Introducing JSON," [Online]. Available: <http://www.json.org/>. [Accessed Januari 2017].
- [8] C. P. Dea, JavaFX 2.0: Introduction by Example, New York: Apress, 2011.
- [9] M. Yener and A. Theedom, Professional Java EE Design Patterns, Indianapolis: John Wiley & Sons, Inc., 2015.
- [10] Warkim and D. I. Senses, "Model Integrasi Sistem dengan Pendekatan Metode Service Oriented Architecture dan Model View Controller pada Pusat Penelitian Perkembangan Iptek Lembaga Ilmu Pengetahuan Indonesia," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 3, no. 1, pp. 84-103, 2017.
- [11] J. L. Weaver, W. Gao, S. Chin, D. Inverson and J. Vos, Pro JavaFX 2: A Definitive Guide to Rich Clients with Java Technology, New York: Apress, 2012.
- [12] T. Kaczanowski, Practical Unit Testing with JUnit and Mockito, Warsaw: Kaczanowscy, 2013.
- [13] "PHP 5 Introduction," Refsnes Data, [Online]. Available: [https://www.w3schools.com/php/php\\_intro.asp](https://www.w3schools.com/php/php_intro.asp). [Accessed September 2017].