

Aplikasi Penjadwalan Berbasis Heuristik untuk Peserta Pelatihan dengan Waktu dan Materi yang Berbeda-Beda

Heuristic-based Scheduling Application for Trainees with Different Timetables and Courses Material

Yustina Sri Suharini^{1*}

Program Studi Informatika Institut Teknologi Indonesia
Jalan Raya Puspiptek Serpong, Kota Tangerang Selatan, Banten 15320, Indonesia
< yustina@iti.ac.id >

Abstrak

Sebagian besar lembaga pelatihan atau balai latihan kerja menggunakan penjadwalan dengan model batch, yang artinya sebuah jadwal digunakan bersama-sama untuk sekelompok orang, tanpa melihat karakteristik masing-masing peserta pelatihan atau ketersediaan waktu mereka. Namun penjadwalan model batch seperti itu belum tentu efektif untuk setiap orang dan belum sesuai dengan prinsip student center learning. Tulisan ini menawarkan alternatif solusi bagi lembaga pelatihan atau balai latihan kerja yang memerlukan penjadwalan dengan keragaman materi pelatihan dan ketersediaan waktu setiap peserta yang berbeda-beda. Solusi berupa pembuatan perangkat lunak aplikasi penjadwalan dengan arsitektur model-view-controller. Perangkat lunak yang dibuat mampu menjadwalkan lebih dari 300 peserta pelatihan dengan jumlah instruktur lebih dari 14 orang dan materi pelatihan berjumlah lebih dari 18 macam dengan tingkat kedalaman yang beragam.

Kata Kunci : penjadwalan pelatihan, slot waktu dan materi berbeda-beda

Abstract

It is common for training institutions or vocational training centers using batch scheduling model, which means that a timetable be used together for a group of people, regardless of their individual needs or their time availabilities. Scheduling in batch model like that was not necessarily effective for every class member and not in accordance with the principle of student center learning. This paper offers an alternative scheduling solution for training institutions or vocational training centers that each participant has special needs in timetable, course interest, and course level. Our solution was implemented by model-view-controller architecture. The software could be used to schedule of more than 300 trainees with more than 14 instructors and more than 18 kinds of training materials.

Keyword : training scheduling, different timetables and course material

1. Pendahuluan

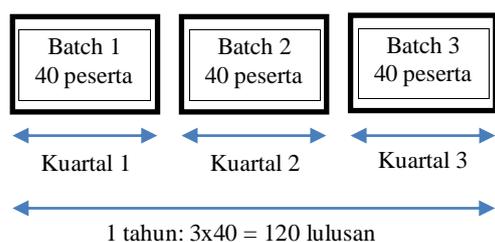
Era globalisasi memungkinkan para calon tenaga kerja mencari pekerjaan secara lebih bebas di perusahaan-perusahaan dalam dan luar negeri. Demikian juga perusahaan-perusahaan yang memerlukan tenaga kerja, dimungkinkan untuk mempunyai lebih banyak pilihan tenaga kerja yang berasal dari berbagai negara. Dalam hal ini kualifikasi atau kompetensi calon tenaga kerja

menjadi faktor penting yang mempertemukan masing-masing calon tersebut dengan perusahaan yang sesuai.

Balai Latihan Kerja (BLK) merupakan lembaga yang ditujukan untuk melatih anggota masyarakat, khususnya mereka yang tergolong kelompok usia produktif, agar mempunyai ketrampilan tertentu sehingga siap terjun ke dunia kerja. Materi yang dilatihkan berupa

pendidikan keterampilan praktis seperti teknik las, teknik kelistrikan, teknik AC (*air conditioning*), teknik sepeda motor, teknik mesin bubut, teknik mesin jahit, teknik penggunaan komputer, dan lainnya. Hingga saat ini terdapat lebih dari 300 BLK milik pemerintah propinsi, kabupaten, dan kota yang tersebar di seluruh Indonesia, di luar BLK yang didirikan oleh swasta dan asing. Pengelolaan yang memadai akan bisa meningkatkan efisiensi dan kinerja BLK, sehingga peran BLK semakin nyata dalam menghasilkan tenaga kerja terampil. Salah satu hal yang bisa ditingkatkan pengelolaannya adalah sistem penjadwalan.

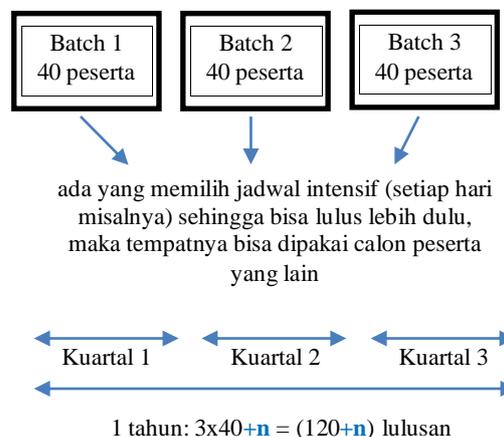
Pada sebagian besar BLK yang ada, materi pelatihan diberikan per paket untuk kurun waktu tertentu dengan jumlah peserta sesuai kapasitas atau daya tampung BLK. Sebagai contoh, pelatihan Komputer Tingkat Dasar, yang materinya berisi Word Processor, Spread Sheet, dan Presentasi, membutuhkan waktu pelatihan selama tiga bulan dengan jumlah peserta 40 orang sesuai ketersediaan jumlah komputer dan instruktur di BLK. Pada akhir masa pelatihan biasanya diadakan ujian untuk mengetahui tingkat ketrampilan yang berhasil dicapai peserta, diikuti dengan penyerahan sertifikat bagi peserta yang berhasil lulus ujian. Setelah itu BLK akan menerima 40 peserta pelatihan komputer untuk *batch* lainnya sampai kurun waktu berikutnya. Begitu seterusnya siklus pelayanan BLK yang jika ada tiga kali pelatihan per tahun, maka BLK tersebut akan menghasilkan maksimal $3 \times 40 = 120$ lulusan Komputer Tingkat Dasar tiap tahun. Demikian juga untuk pelatihan-pelatihan bidang lain yang diselenggarakan di BLK.



Gambar 1. Penjadwalan Cara *Batch*

Penjadwalan pelatihan dengan cara penanganan per *batch* seperti itu relatif sederhana sehingga mudah dilaksanakan, tetapi belum tentu optimal. Misalnya jika ada peserta-peserta yang mengundurkan diri sebelum selesai pelatihan, mestinya ada tempat-tempat kosong namun tidak terpakai sampai pergantian *batch* berikutnya. Contoh lainnya, jika ada peserta yang punya kemampuan lebih baik atau lebih terampil dibanding teman-teman di *batch* yang sama, dia tetap harus menunggu para peserta lain untuk

mengikuti ujian bersama-sama yang biasanya dilaksanakan di akhir masa pelatihan. Padahal jika seorang peserta bisa lulus tanpa harus menunggu, maka tempatnya akan bisa dipakai oleh orang lain yang sedang menunggu antrian untuk mulai ikut pelatihan.



Gambar 2. Penjadwalan dengan Waktu dan Materi yang Berbeda-Beda untuk Tiap Peserta

Pada tulisan ini diusulkan sebuah alternatif solusi bagi BLK yang membutuhkan jadwal yang berbeda-beda sesuai ketersediaan waktu peserta, materi yang dibutuhkan oleh setiap peserta, dan ketersediaan instruktur. Usulan yang dimaksud berupa *software* aplikasi yang dapat digunakan untuk membantu penjadwalan BLK secara dinamis. Maksud dinamis di sini adalah keleluasaan setiap peserta untuk menentukan jadwal pelatihannya sendiri sesuai dengan pilihan waktu dan kemampuan masing-masing, termasuk jadwal ujian yang berbeda-beda antara peserta yang satu dengan peserta yang lain. Misalnya seorang calon peserta yang bernama Ana mulai pelatihan pada tanggal 06 Februari 2017, dengan frekuensi seminggu 3 kali yaitu setiap hari Senin, Selasa, Rabu jam 14:00-16:00 (ditentukan oleh Ana sendiri dengan cara mengisi formulir pada saat mendaftar). Sementara itu Budi yang juga calon peserta mulai pelatihan pada tanggal 27 Februari 2017, dengan frekuensi seminggu sekali tiap hari Kamis jam 08:00-16:00 (ditentukan oleh Budi sendiri saat mengisi formulir pendaftaran). Kemudian Ana dan Budi mengikuti pelatihan sesuai jadwal dan materi serta capaian kualifikasi masing-masing. Melalui sistem penjadwalan yang dinamis seperti ini, produktivitas BLK dalam menghasilkan lulusan bisa meningkat, sehingga percepatan jumlah tenaga terampil semakin nyata.

Tulisan ini disusun dengan urutan berikut. Pertama-tama disajikan dasar teori, diikuti oleh metodologi yang digunakan. Pada metodologi

diuraikan langkah-langkah pembuatan *software*. Bagian berikutnya adalah hasil implementasi dan pembahasan. Bagian terakhir adalah kesimpulan dan rencana pengembangan.

2. Teori Dasar

Persoalan penjadwalan telah dikenal luas sebagai persoalan dengan kompleksitas tinggi [1]. Oleh karena itu para peneliti masih berusaha keras menciptakan berbagai macam pendekatan untuk menyelesaikannya. Dalam beberapa dekade terakhir, sudah banyak algoritma dikembangkan dan diujicobakan pada persoalan nyata. Beberapa pendekatan yang diusulkan antara lain pendekatan yang berdasarkan sifat alamiah gen makhluk hidup, atau dikenal dengan *genetic algorithm* (GA). Pendekatan lain yang juga populer adalah pendekatan yang meniru perilaku kawanan binatang yaitu *particle swarm optimization* (PSO) [2] [3], *Ant Colony Optimization* (ACO) [4] [5], dan *Artificial Bee Colony* (ABC) [6]. Ada juga pendekatan lain yaitu algoritma *Teaching-Learning Based Optimization* (TLBO) yang diusulkan dalam berbagai literatur sebagai algoritma meta-heuristik terbaru berorientasi pada populasi [1].

Dalam perkembangannya, pendekatan-pendekatan tersebut telah mengalami perubahan dari para peneliti saat ini, misalnya GA sudah ditingkatkan kemampuannya dengan sifat *bacteria foraging* menjadi Genetic Algorithms Bacteria Foraging Optimization (GABFO) [7]. Demikian pula PSO dikembangkan lebih lanjut dengan penambahan kemampuan logika fuzzy, menjadi Fuzzy PSO maupun *Revised Discrete Particle Swarm Optimization* (RDPSO) [8].

Sementara itu banyak juga pendekatan lain untuk penjadwalan pada tingkat sistem, misalnya penjadwalan prosesor yang terdiri atas ribuan *node* pada jaringan komputer, serta sistem TDMA, *hop*, *wireless sensor network*. Kesemuanya itu termasuk persoalan optimasi NP-complete sehingga membutuhkan algoritma penjadwalan yang cukup memadai sebagai pendekatan solusi [9]. Beberapa contoh algoritma yang dikembangkan dalam tingkat sistem antara lain *Min-Min Average* (MMA) [10], *Own Criticality Based Priority* (OCBP) [11], dan lanjutan OCBP yaitu *Criticality Based Earliest Deadline First* (CBEDF) [11], *Rate-Based Scheduling Algorithms* [12], *Maximal Scheduling* [13], dan *Wireless Scheduling Algorithms* [13].

Penjadwalan dapat juga dilakukan dengan teori himpunan dan relasi-relasi antar himpunan. Item-item yang mempunyai karakteristik tertentu dapat dikelompokkan menjadi satu kelompok atau satu himpunan, seperti siswa, slot waktu, materi pelatihan, dan instruktur. Jika ada item-

item yang lebih spesifik maka bisa dimasukkan ke dalam sub himpunan, misalnya slot-slot waktu yang berada di hari-hari tertentu, level atau kedalaman materi pelatihan yang hendak diikuti oleh calon siswa, instruktur yang menjadi preferensi atau pilihan siswa di antara sejumlah instruktur yang tersedia. Begitu pula jika seorang siswa memilih slot waktu tertentu maka akan terjadi relasi pada sebuah item di himpunan calon siswa dengan sebuah item di slot tertentu. Bisa saja satu siswa memilih dua atau tiga slot dalam seminggu, berarti ada tiga relasi terjadi dari sebuah item di himpunan calon siswa dengan tiga item dalam himpunan slot.

Menurut Miguel L Lerma [14], himpunan adalah kumpulan item. Misalnya himpunan D terdiri atas tiga item x, y, z, maka himpunan D dinyatakan sebagai:

$$D = \{ x, y, z \} \dots\dots\dots (1)$$

Selain dengan menyebutkan anggota-anggotanya, sebuah himpunan juga bisa dinyatakan dengan pernyataan atau formula yang mendefinisikan karakteristik anggota-anggotanya [15].

Anggota himpunan disebut sebagai elemen himpunan dan dituliskan:

$$x, y, z \in D$$

yang jika ditulis masing-masing :

$$x \in D$$

$$y \in D$$

$$z \in D$$

artinya x, y, dan z ketiganya anggota himpunan D.

Himpunan yang merupakan bagian dari himpunan lain disebut himpunan bagian (*sub set*) dan ditulis $E \subset D$ jika himpunan E merupakan bagian dari himpunan D. Contoh himpunan bagian yang ada pada penjadwalan misalnya slot waktu untuk yang ada di hari Senin merupakan himpunan bagian dari himpunan semua slot yang telah ditentukan oleh lembaga pelatihan.

Jika $E = \{ x, y \}$ maka $E \subset D$.

Jika $F = \{ x, y, z \}$ maka $E \subset F$ karena $F = D$.

Irisan atau *intersection* dua himpunan yang ditulis $E \cap F$ ialah himpunan yang anggota-anggotanya merupakan anggota himpunan E sekaligus anggota himpunan F. Irisan pada kasus penjadwalan misalnya seorang instruktur yang bisa mengajar AutoCAD sekaligus mengajar Bahasa Inggris. Irisan bisa terjadi pada lebih dari dua himpunan. Pada contoh di atas,

$$E \cap F = \{ x, y \}$$

$$E \cap D = \{ x, y \}$$

$$F \cap D = \{ x, y, z \}$$

$$E \cap F \cap D = \{ x, y \}$$

Gabungan (*union*) dua himpunan A dan B ditulis $A \cup B$, merupakan penggabungan anggota A dan anggota B. Contoh gabungan pada kasus penjadwalan misalnya daftar semua siswa yang mengikuti pelatihan teknik reparasi sepeda motor, pelatihan las, pelatihan MS Word, dan pelatihan-pelatihan lain yang diselenggarakan oleh lembaga pelatihan.

Jika $A = \{ 1, 2, 3 \}$ dan $B = \{ k, l \}$
maka $A \cup B = \{ 1, 2, 3, k, l \}$

Relasi antara anggota himpunan G dan H ditulis aRd atau Rad dengan $a \in G$ dan $d \in H$ [16]. Relasi dapat juga dinyatakan dengan menyebutkan setiap hubungan antara anggota himpunan G dan anggota himpunan H, misalnya

$$R = \{ \langle a,d \rangle, \langle a,e \rangle, \langle b,c \rangle \}$$

pada semesta pembicaraan $\{ a,b \} \times \{ c,d,e \}$

Komplemen relasi didefinisikan sebagai

$$R' = (G \times H) - R \dots\dots\dots (2)$$

yang untuk contoh di atas

$$R' = \{ \langle a,c \rangle, \langle b,d \rangle, \langle b,e \rangle \} [16]$$

Inverse untuk relasi $R \subseteq G \times H$ didefinisikan sebagai $R^{-1} \subseteq H \times G$ dengan

$$R^{-1} = \{ \langle i,j \rangle \mid \langle j,i \rangle \in R \} \dots\dots\dots (3)$$

sehingga untuk contoh di atas didapat

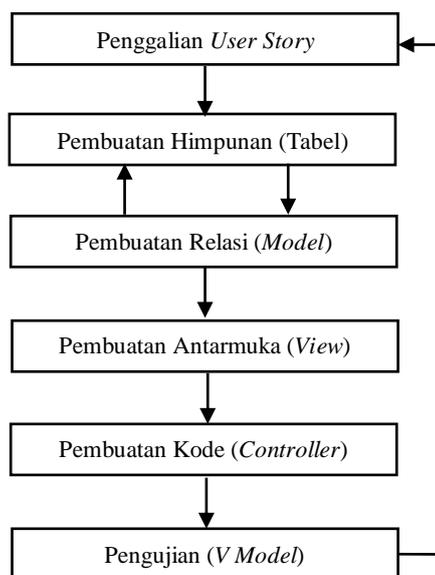
$$R^{-1} = \{ \langle d,a \rangle, \langle e,a \rangle, \langle c,b \rangle \}$$
 dan dari definisi dan contoh jelaslah bahwa $(R^{-1})^{-1} = R$.

Relasi pada kasus penjadwalan terjadi tidak hanya atas himpunan-himpunan melainkan juga atas sub-sub himpunan yang terlibat dalam sistem. Relasi-relasi ini dinamis dalam arti bisa terjadi bisa tidak, sesuai pilihan calon siswa di awal masa pelatihan. Hal-hal yang bisa digunakan sebagai batasan adalah banyaknya slot waktu yang tersedia di masing-masing hari, jumlah instruktur, kompetensi tiap instruktur, kapasitas ruang dan juga kapasitas alat untuk tiap slot waktu pelatihan.

3. Metodologi

Software aplikasi penjadwalan ini dibuat menggunakan arsitektur *model-view-controller*.

[1] Langkah pertama adalah penggalan serta penentuan kebutuhan pengguna (*user requirement / user story*). Pada langkah ini *customer* berperan besar sebagai penyedia *user requirement* atau *user story*. Oleh karena itu komunikasi dengan customer sangat penting. Keberhasilan komunikasi dengan *customer* sangat berpengaruh pada keberhasilan penggalan kebutuhan pengguna. Dengan kata lain keberhasilan penggalan *user requirement* tergantung pada ketepatan komunikasi dengan *customer* sehingga langkah ini juga disebut sebagai *customer communication*.



Gambar 3. Langkah-Langkah Pembuatan *Software*

- [2] Setelah penggalan *user requirement*, langkah kedua adalah pendefinisian himpunan dan keanggotaan himpunan atas item-item yang diperlukan dalam penjadwalan.
- [3] Langkah ketiga adalah penentuan relasi atau hubungan yang bisa terjadi antar himpunan maupun antar sub himpunan. Pada langkah ini dimungkinkan untuk bisa kembali ke langkah kedua apabila revisi himpunan dan sub himpunan diperlukan.
- [4] Langkah keempat berupa pembuatan antarmuka pengguna (*view*). Di langkah ini *customer* berperan sangat intensif dalam penentuan *view* yang paling sesuai untuk kebutuhan penjadwalan.
- [5] Langkah kelima adalah pembangunan kode program (*source code*) atau *controller* yang sesuai dengan *view* yang sudah ditentukan.
- [6] Seiring dengan penulisan kode program di langkah kelima pada sebuah modul, selalu diiringi dengan pengujian unit. Tujuan dari pengujian unit ini untuk memastikan modul sudah benar (*correctness*). Unit-unit yang berkolaborasi melakukan sebuah fitur diuji dengan pengujian integrasi. Setelah semua modul selesai dibuat maka diinstal atau di-*deploy* dan diuji di lingkungan *customer* (pengujian sistem). Setelah pengujian sistem maka dilanjutkan dengan pengujian penerimaan oleh *user* (*user acceptance testing*). Secara lengkap rangkaian pengujian yang dimulai dari pengujian unit (*unit testing*), pengujian

integrasi (*integration testing*), pengujian sistem (*system testing*), hingga pengujian penerimaan user (*user acceptance testing*) disebut sebagai V-Model atau V-Testing.

Langkah-langkah pembuatan *software* aplikasi penjadwalan disajikan pada Gambar 1. Pelibatan *customer* selama proses pembuatan *software* menjadi ciri khas metodologi yang digunakan. Meskipun tidak secara eksplisit kelihatan namun peran *customer* dalam penentuan *user story* dan *view* sangat besar.

4. Hasil dan Pembahasan

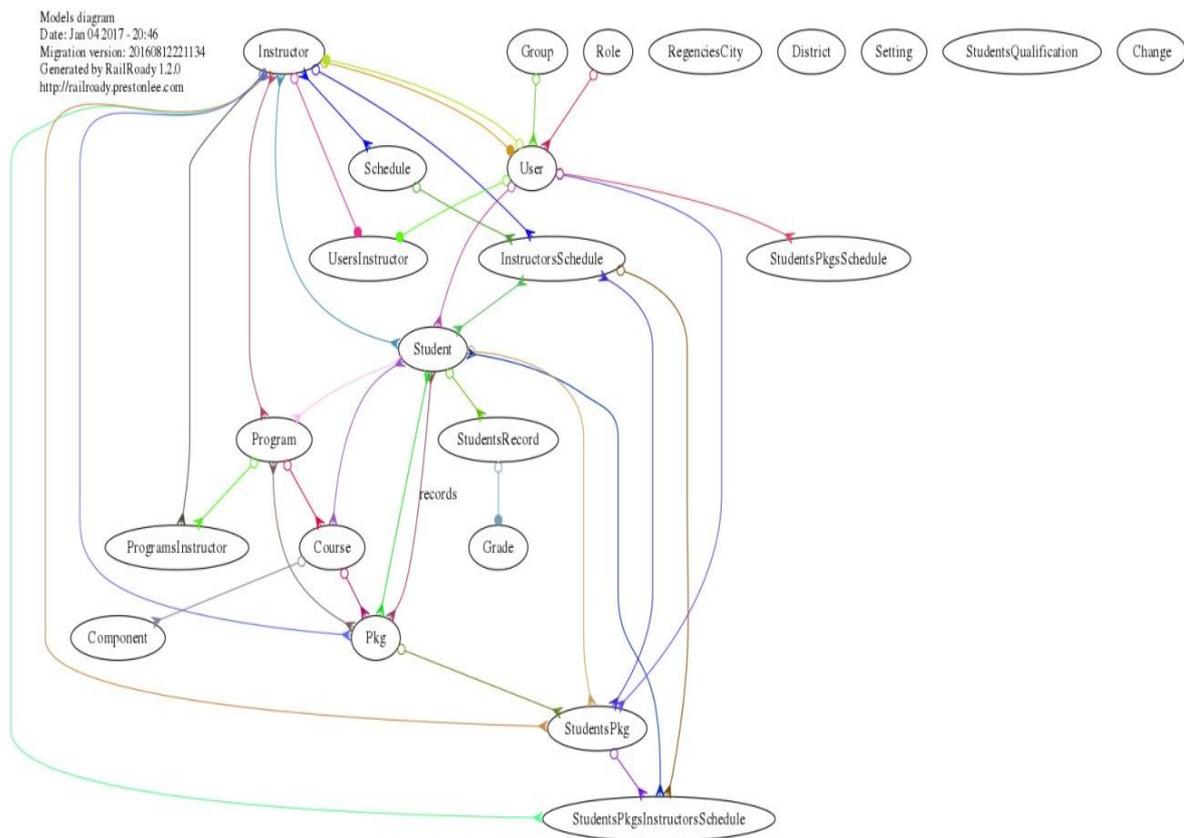
Implementasi menggunakan arsitektur *model-view-controller* memberikan hasil berikut. Himpunan dan irisan himpunan item-item yang terlibat serta relasi yang terjadi di antara mereka diimplementasikan sebagai *model*. Sedangkan cara menggunakan software dari sudut pandang

user ditentukan berdasarkan *user story* atau *user requirement*, yang diimplementasi sebagai *view*. *Controller* merupakan kode program penghubung *model* dan *view*.

Tabel 1. Himpunan dan Irisan Himpunan

User	UserInstructor
Instructor	InstructorSchedule
Student	StudentPkg
Program	ProgramInstructor
Schedule	StudentPkgSchedule
Pkg	StudentPkgInstructorSchedule
Record	StudentRecord

Pertama-tama, daftar himpunan dan sub himpunan disajikan pada Tabel 1. Gambar 4 adalah *model* yang dibuat, berisi himpunan-himpunan dan sub-sub himpunan item-item yang terlibat dalam penjadwalan serta relasi-relasi yang terjadi.

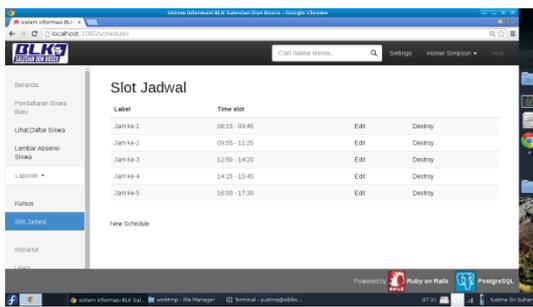


Gambar 4. Model untuk Aplikasi Penjadwalan

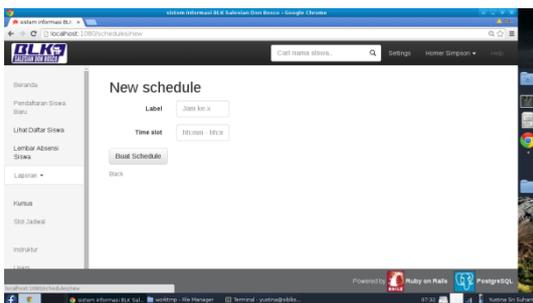
Gambar 5 sampai 14 merupakan beberapa contoh *view* yang sudah dibuat untuk aplikasi penjadwalan di Balai Latihan Kerja Salesian Don Bosco (BLK SDB) yang berlokasi di Tigaraksa

Kabupaten Tangerang. Sedangkan Tabel 2 adalah daftar *method* atau *controller* yang dibuat untuk menghubungkan *model* dan *view* yang telah ditentukan berdasarkan *user story*.

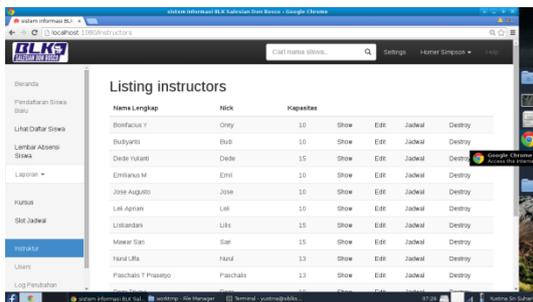
Aplikasi Penjadwalan Berbasis Heuristik untuk Peserta Pelatihan dengan Waktu dan Materi yang Berbeda-Beda
Yustina Sri Suharini



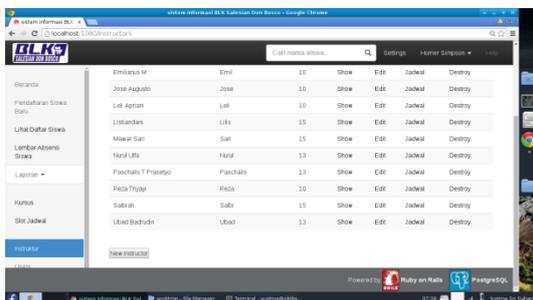
Gambar 5. Pengelolaan Slot Waktu



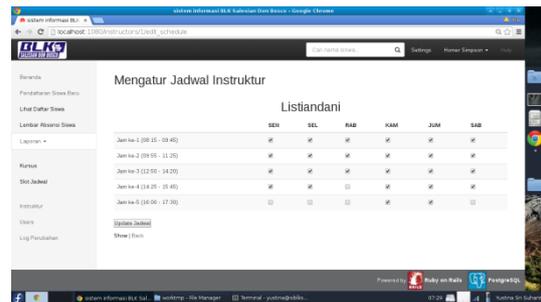
Gambar 6. Membuat Slot Baru



Gambar 7. Daftar Instruktur



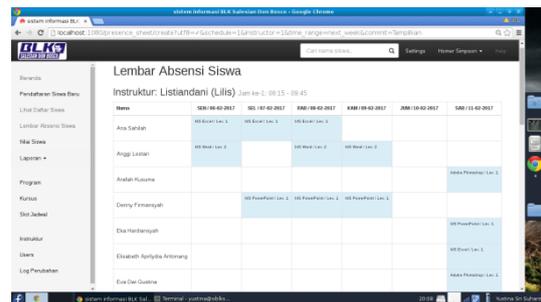
Gambar 8. Daftar Instruktur (Lanjutan)



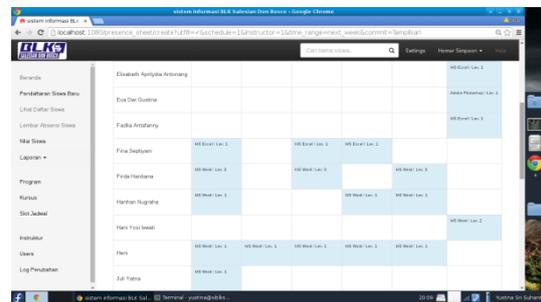
Gambar 9. Form Kesanggupan Instruktur 1



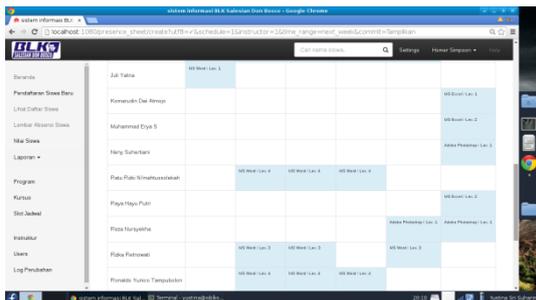
Gambar 10. Form Kesanggupan Instruktur 2



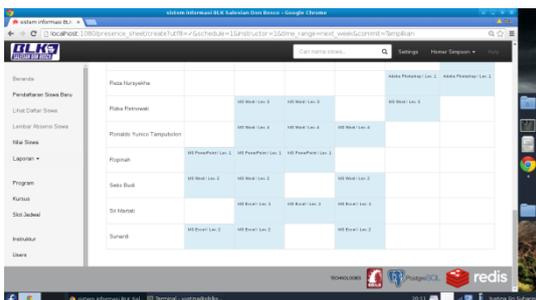
Gambar 11. Contoh Daftar Hadir Siswa yang memilih slot dan Instruktur 1



Gambar 12. Contoh Daftar Hadir Siswa yang memilih slot dan Instruktur 1 (halaman 2)



Gambar 13. Contoh Daftar Hadir Siswa yang memilih slot dan Instruktur 1 (halaman 3)



Gambar 14. Contoh Daftar Hadir Siswa yang memilih slot dan Instruktur 1 (halaman 4)

Tabel 2. Controller Aplikasi Penjadwalan

ReportController	HelpController
GradesController	InstructorScheduleController
GroupsController	PresenceSheetController
StudentsController	PasswordResetController
ProgramsController	PkgsController
SchedulesController	StudentScheduleController
WelcomeController	StudentRecordController
SessionsController	ProgramInstructorsController
SettingsController	InstructorsController
UsersController	ChangesController
ProfileController	ApplicationsController



Gambar 15. Jumlah Siswa Aktif Januari 2017

Software yang dibuat telah digunakan di Balai Latihan Kerja Salesian Don Bosco (BLK SDB) yang berlokasi di Tigaraksa Kabupaten Tangerang. Hingga saat ini tercatat 1865 siswa yang penanganannya telah dibantu oleh

software dimaksud. Dalam satu periode ada lebih dari 300 siswa aktif, sebagai contoh pada bulan Januari 2017 terdapat 376 siswa aktif yang terdiri atas 133 siswa laki-laki dan 243 siswa perempuan seperti ditunjukkan pada Gambar 15.

Jumlah pelatihan hingga sekarang tercatat ada sekitar 25 seperti ditunjukkan pada Tabel 3.

Tabel 3. Daftar Pelatihan

MS Word	AC
MS Excel	Las Dasar
MS PowerPoint	Mekanik Bubut
MS Access	Sepeda Motor
Corel Draw	Sepeda Motor - Basic
Adobe Photoshop	Sepeda Motor - Electrical
AutoCAD 2D	Sepeda Motor - Overhaul
AutoCAD 3D	Sepeda Motor - Injection
MS Word	Elementary I
Web Design - HTML	Elementary II
Web Design - CSS	Intermediate
Web Design - JQuery	Advance
Listrik Rumah Tangga & Industri	

Terdapat lima slot waktu tiap hari yang bisa dipilih oleh para calon siswa, seperti pada Tabel 4.

Tabel 4. Slot Waktu per Hari

Jam ke-1	08:15 - 09:45
Jam ke-2	09:55 - 11:25
Jam ke-3	12:50 - 14:20
Jam ke-4	14:25 - 15:45
Jam ke-5	16:00 - 17:30

Hingga saat tulisan ini dibuat, terdapat 15 user tercatat di dalam sistem, 2 user termasuk dalam kategori *system administrator*, 3 user *staff* atau sekretariat, 10 user termasuk instruktur yang sudah senior. Jumlah instruktur sendiri tercatat ada 14, jadi tidak semua instruktur mempunyai akses ke sistem penjadwalan ini. Oleh karena itu peran *staff* kesekretariatan masih dominan dalam mengentri form kesanggupan bagi para instruktur yang tidak bisa atau tidak diijinkan mengakses sistem secara langsung.

5. Kesimpulan

Software aplikasi penjadwalan untuk lembaga pelatihan dengan alokasi waktu dan materi yang berbeda-beda tiap peserta dengan pendekatan heuristik berupa himpunan dan relasi telah berhasil dibuat. Aplikasi ini telah digunakan oleh Balai Latihan Kerja Salesian Don Bosco (BLK SDB) yang berlokasi di Tigaraksa Kabupaten Tangerang sejak akhir 2014 dan hingga sekarang telah alami banyak sekali perubahan berupa penambahan modul-modul sesuai dinamika perubahan yang terjadi di BLK SDB.

Ucapan Terima Kasih

Terima kasih kepada Yth. Direktur Balai Latihan Kerja Salesian Don Bosco Tigaraksa Kabupaten Tangerang, Pastur Paschalis Tri Prasetyo, yang telah mengizinkan kami membuat aplikasi penjadwalan ini dan menggunakannya untuk mengelola jadwal instruktur dan jadwal siswa BLK SDB hingga sekarang.

Terima kasih juga kepada Bp Agustinus Edwin Pratomo Adiwino yang telah menjadi mentor sekaligus pemberi inspirasi.

Daftar Pustaka

- [1] Baykasoğlu, Adil; Hamzadayi, Alper; Köse, Simge Yelkenci, 2014, “*Testing the performance of teaching-learning based optimization (TLBO) algorithm on combinatorial problems*”, Journal Information Sciences Volume 276 Issue C, Pages 204-218.
- [2] Lin, Tsung-Lieh; Horng, Shi-Jinn; Kao, Tzong-Wann; Chen, Yuan-Hsin; Run, Ray-Shine; Lai, Jui-Lin; Kuo, I-Hong; 2010, “*An Efficient Job-Shop Scheduling Algorithm Based On Particle Swarm Optimization*”, Journal Expert Systems with Applications, Volume 37 Issue 3, March, 2010 Pages 2629-2636.
- [3] Kader, Rehab F. Abdel; “*Particle Swarm Optimization for Constrained Instruction Scheduling*”, Hindawi Publishing Corporation, VLSI Design Volume 2008.
- [4] Tavares, R.F.; Filho, M. Godinho; 2013, “*Literature Review Regarding Ant Colony Optimization Applied To Scheduling Problems: Guidelines For Implementation And Directions For Future Research*”, Journal Engineering Applications of Artificial Intelligence, Volume 26 Issue 1, Pages 150-161.
- [5] Rossi, Andrea; Lanzetta, Michele; 2014, “*Native Metaheuristics for Non-Permutation Flowshop Scheduling*”, Journal of Intelligent Manufacturing, Springer-Verlag Volume 25 Issue 6, Pages 1221-1233.
- [6] Banharsakun, Anan; Sirinaovakul, Booncharoen; Achalakul, Tiranee; 2012, “*Job Shop Scheduling with the Best-so-far ABC*”, Journal Engineering Applications of Artificial Intelligence, Volume 25 Issue 3, April, 2012, Pages 583-593.
- [7] Prakash, Shiv; Vidyarthi, Deo Prakash; 2014, “*A Hybrid GABFO Scheduling for Optimal Makespan in Computational Grid*”, Journal International Journal of Applied Evolutionary Computation IGI Publishing Hershey Volume 5 Issue 3, July 2014, Pages 57-83.
- [8] Chen, Chun-Lung; Huang, Shin-Ying; Tzeng, Yeu-Ruey; Chen, Chuen-Lung; 2014, “*A Revised Discrete Particle Swarm Optimization Algorithm for Permutation Flow-Shop Scheduling Problem*”, Journal Soft Computing - A Fusion of Foundations, Methodologies and Applications, Springer Volume 18 Issue 11, November 2014, Pages 2271-2282.
- [9] Ergen, Sinem Coleri; Varaiya, Pravin; 2010, “*TDMA Scheduling Algorithms for Wireless Sensor Network*”, Springer Wireless Network 16 page 985-997.
- [10] Liu, Ke; Chen, jinjun; Jin, Hai; Yang, Yun; “*A Min-Min Average Algorithm for Scheduling Transaction-Intensive Grid Workflows*”, Proc. 7th Australasian Symposium on Grid Computing and e-Research (AusGrid 2009), Wellington, New Zealand, 2009.
- [11] Park, Taeju; Kim, Soontae; “*Dynamic Scheduling Algorithm and Its Schedulability Analysis for Certifiable Dual-Criticality Systems*”, EMSOFT’11, October 9–14, 2011, Taipei, Taiwan, 2011.
- [12] Huang, Po-Kai; Lin, Xiaojun; Chun, Chih; 2013, “*A Low-Complexity Congestion Control And Scheduling Algorithm For Multihop Wireless Networks With Order-Optimal Per-Flow Delay*”, IEEE / ACM Transactions on Networking, Vol. 21, No. 2, Page 1846-1859.
- [13] Gupta, Abhinav; Lin, Xiaojun; Srikant, R.; 2009, “*Low Complexity Distributed Scheduling Algorithms for Wireless Networks*”, IEEE / ACM Transaction on Networking Vol 17 No. 6.
- [14] Lerma, Miguel A., 2005, Mathematical Foundation of Computer Science, Set Theory, Chapter 2: “*Set, Function, Relation*”, (Online),

(<http://www.math.northwestern.edu/~mlerma/courses/cs310-05s/notes/dm-sets.pdf>
diakses Desember 2016)

- [15] Partee, Barbara H., Meulen, Alice Ter, and Wall, Robert. 1990. *Mathematical Methods in Linguistics*, Dordrecht: Kluwer
- [16] Ling 310, adapted from UMass Ling 409, Partee lecture notes March 1, 2006, page 1-10 (online)
http://people.umass.edu/partee/NZ_2006/Set%20Theory%20Basics.pdf (diakses Desember 2016)