

Pengembangan Sistem Student Relationship Management di Universitas Surabaya

Kristian Tanuwijaya¹, Liliana^{2*}, Daniel Soesanto³

^{1,2*,3} Program Studi Teknik Informatika, Universitas Surabaya, Surabaya, Jawa Timur
Email: ¹s160416024@student.ubaya.ac.id, ^{2*}lili@staff.ubaya.ac.id, ³daniel.soesanto@staff.ubaya.ac.id

(Naskah masuk: 24 Apr 2020, direvisi: 8 Jun 2020, diterima: 23 Jun 2020)

Abstrak

Industri komersial saat ini menerapkan teknik *Customer Relationship Management* (CRM) untuk mendapat berbagai keuntungan seperti menyediakan informasi pada pelanggan, meningkatkan loyalitas dan kepercayaan pelanggan, serta mempelajari perilaku pelanggan. Pendidikan dapat dipandang sebagai industri bidang jasa. Pada model tersebut, peran siswa dapat dipadankan dengan konsumen. Karenanya, ada peluang untuk menerapkan CRM dalam dunia pendidikan. Oleh karena itu, penelitian dilakukan untuk mengembangkan aplikasi *student relationship management* untuk Universitas Surabaya yang mencakup rekomendasi beasiswa dan acara, pengingat *event*, dan perekrutan panitia. Aplikasi yang dikembangkan menggunakan platform web dan pembuatan rekomendasi dilakukan dengan algoritma *machine learning* yaitu *random forest*, *deep learning*, dan *stacked ensemble*. Berdasarkan hasil uji coba dan validasi, aplikasi tersebut dapat membantu mahasiswa mengetahui informasi seperti beasiswa yang dapat diperoleh dan kegiatan yang dapat diikuti. Dengan demikian, ketika kepuasan mahasiswa terhadap layanan yang diberikan oleh universitas meningkat, maka hubungan baik antara penyedia jasa, dalam hal ini universitas, dan konsumennya, dalam hal ini mahasiswa, dapat dijaga.

Kata Kunci: Customer Relationship Management, Machine Learning, Sistem Rekomendasi, Aplikasi Web

The Development of Student Relationship Management in University of Surabaya

Abstract

Modern commercial industries have been utilizing Customer Relationship Management (CRM) to get benefits such as providing information to its customer, increasing customer loyalty, and providing insight on customer behavior. Education can be perceived as service industry, therefore education may also get benefit from implementing customer relationship management with student as its customer. This research aims to develop customized CRM application for University of Surabaya. The application's scope includes scholarship and event recommendation, event notification, and committee recruitment system. The application would run on web platform and its recommendation would use machine learning algorithm such as distributed random forest algorithm, deep learning, and stacked ensemble learning. Based on testing and validation, the application has been confirmed to help students obtain scholarship and event information. Thus, when the student satisfaction with the services provided by the university increase, good relations between service providers, in this case the university, and its consumers, in this case students, can be maintained.

Keywords: *customer relationship management, machine learning, recommendation system, web application.*

I. PENDAHULUAN

Industri komersial saat ini telah menerapkan berbagai macam teknik *Customer Relationship Management* (CRM) untuk mendapat berbagai keuntungan [1][2][3]. Pada

pendidikan sebagai industri, peran siswa dapat dipadankan dengan peran konsumen. Oleh karena itu, ada peluang untuk menerapkan CRM dalam dunia pendidikan. Salah satu contoh penerapan CRM dalam dunia pendidikan tersebut adalah

penelitian studi kasus tentang perkembangan penerapan CRM pada *The North Carolina Community College System* (NCCCS) dari tahun 1997 hingga 2006 yang diberi nama *student system* [4]. Dalam penelitian yang dilakukan oleh Seeman, penerapan CRM dalam sistem informasi akademis terbatas pada penyajian data yang terpusat dan akses mahasiswa terhadap informasi akademis yang fleksibel. Penelitian selanjutnya terkait CRM dalam dunia pendidikan tinggi dilakukan pada tahun 2016 [5]. Dalam penelitian ini, penerapan CRM dikelompokkan menjadi enam kelompok, seperti penyediaan *feedback* oleh siswa, kustomisasi layanan yang dibutuhkan oleh mahasiswa, serta penyediaan informasi lowongan kerja untuk mahasiswa yang membutuhkan. Dasar dari penelitian ini adalah wawancara yang dilakukan terhadap beberapa siswa. Selain itu, penelitian di tahun yang sama dilakukan pada bidang yang sama, dan penelitian ini menyatakan bahwa CRM dibutuhkan oleh pendidikan tinggi untuk meningkatkan kepuasan siswanya [6].

Secara umum penelitian-penelitian sebelumnya telah membuktikan secara kualitatif bahwa penerapan sistem informasi CRM pada pendidikan tinggi dapat mempertahankan, meningkatkan loyalitas mahasiswa sebagai pelanggan, membantu tenaga pendukung akademik, dan menjadi *competitive advantage*. Namun pada penelitian-penelitian sebelumnya semuanya masih sebatas asumsi dan hasil survei kualitatif, belum ditemukan implementasi nyata dari sistem yang dirancang.

Kontribusi dari penelitian ini adalah mengimplementasikan desain rancangan CRM pada sebuah aplikasi yang dapat memahami kebutuhan informasi mahasiswa dan kemudian menyajikan informasi tersebut. Pendekatan ini berkebalikan dari norma umum dimana mahasiswa harus mencari informasi sendiri melalui papan pengumuman, *official website* atau bertanya ke unit yang menangani. Sedangkan pada aplikasi ini, Universitas Surabaya secara proaktif dapat melakukan pendekatan langsung pada mahasiswa yang dianggap layak untuk menerima beasiswa tersebut, berdasar syarat untuk masing-masing beasiswa. Dengan perlakuan tersebut, diharapkan mahasiswa akan merasa diperhatikan sehingga dapat meningkatkan loyalitasnya sebagai civitas Universitas Surabaya. Dari empat tugas jenis utama CRM (*strategic, operational, analytical, collaborative*) [7], aplikasi yang dikembangkan termasuk *operational* dan *analytical* CRM.

Analisis permasalahan menunjukkan bahwa ada empat jenis masalah penyebaran informasi yang dapat ditangani oleh aplikasi CRM yang akan dikembangkan yaitu pemberitahuan dan rekomendasi beasiswa, pengingat dan rekomendasi kegiatan, pengingat jadwal membayar uang kuliah, dan perekrutan panitia. Pemberitahuan dan rekomendasi beasiswa berfungsi untuk menyebarkan informasi beasiswa yang disediakan Universitas Surabaya pada mahasiswa maupun calon mahasiswa, dan memberi rekomendasi beasiswa pada mahasiswa. Pengingat dan rekomendasi kegiatan berfungsi untuk menampilkan data kegiatan yang ada pada Universitas Surabaya dan memberi pengingat apabila mahasiswa memilih berpartisipasi di dalamnya. Sedangkan perekrutan panitia adalah fitur perekrutan yang mirip dengan aplikasi *job vacancy*

tetapi digunakan untuk merekrut panitia untuk kegiatan Universitas Surabaya. Dalam penelitian kali ini, fokus ditekankan pada sistem rekomendasi pemberian beasiswa.

Pengembangan aplikasi dibatasi dengan *web application* sebagai platform, pengingat yang menggunakan *Google Calendar*, dan sistem rekomendasi yang menggunakan *automated machine learning*. Pengingat *Google Calendar* dikirimkan pada akun *Google Business Suite (G-Suite)* milik Universitas Surabaya yang lebih sering disebut Gooaya. Sistem rekomendasi dibatasi menggunakan tugas *multi-class classification* dengan algoritma *random forest* [8], *deep learning* [9], dan *stacked ensemble* [10]. Selain itu rekomendasi harus dapat diberikan secara manual, yaitu syarat-syarat rekomendasi dimasukkan oleh administrator sistem.

REST (*Representational State Transfer*) API adalah API (*Application Programming Interface*) yang dijalankan pada arsitektur web dengan HTTP (*Hypertext Transfer Protocol*). Implementasi RESTful API yang diterapkan dengan baik dapat meningkatkan kinerja sistem, *scalability*, kesederhanaan, modularitas, portabilitas, reliabilitas, dan visibilitas [11]. Untuk mengintegrasikan aplikasi dengan *Google Calendar*, digunakan layanan *Google Calendar REST API*. Layanan tersebut merupakan implementasi REST API untuk mengakses *calendar resource* pada *Google Calendar* yang menggunakan otentikasi dan otorisasi dengan *Oauth2*.

Agar aplikasi dapat memberikan rekomendasi pada mahasiswa, seperti rekomendasi beasiswa dan kepanitiaan, diperlukan sebuah sistem rekomendasi. Teknik sistem rekomendasi menggunakan metode *machine learning*, seperti *classification* dan *clustering* terhadap data matriks interaksi pengguna dengan konten [12]. Istilah *machine learning* itu sendiri mengacu pada proses membuat model dari hasil mempelajari data yang kemudian digunakan untuk membuat prediksi [13]. Pada kasus ini data sumber pelajaran (*training data set*) adalah data mahasiswa yang sudah ada pada sistem terintegrasi myubaya.ac.id. Contoh data yang digunakan adalah penerima beasiswa, panitia kegiatan, daftar peserta berbagai kegiatan di masa lalu, dan prediksi yang dibuat adalah siapa yang akan menerima beasiswa, panitia yang sesuai untuk direkrut, dan kegiatan yang sesuai untuk mahasiswa di masa depan. Proses ini disebut sebagai *classification*.

II. METODE PENELITIAN

Metode penelitian yang digunakan pada pengembangan aplikasi *student relationship management* ini dijelaskan dalam langkah-langkah berikut.

1. Perencanaan dan Persiapan

Pada tahap ini, peneliti mencari permasalahan dan kebutuhan sistem secara umum pada Universitas Surabaya. Peneliti juga melakukan studi pustaka untuk memperdalam pemahaman teori dan penelitian sebelumnya.

2. Analisis Sistem

Peneliti mengumpulkan dan menganalisis kebutuhan sistem. Analisis awal dilakukan melalui survei terhadap mahasiswa Universitas Surabaya dari berbagai jurusan, serta dosen pengajarnya. Selain itu, peneliti juga mengumpulkan informasi pendukung seperti kebijakan dari Universitas Surabaya. Selanjutnya peneliti menentukan fitur-fitur yang akan dibuat sesuai analisis kebutuhan sistem.

3. Desain Sistem

Desain data dilakukan dengan notasi *Entity Relationship Diagram* (ERD). Desain proses dilakukan dengan notasi *Business Process Model and Notation* (BPMN). Desain *user interface* dilakukan dengan *high-fidelity prototype*. Selain itu, dilakukan desain proses untuk *machine learning* yang akan diterapkan.

4. Implementasi

Implementasi dilakukan dengan membangun aplikasi *student relationship management* dari desain yang telah dikerjakan. Implementasi dilakukan dengan bahasa pemrograman PHP dengan *framework Laravel* untuk pembuatan *web application*, bahasa pemrograman Python dengan library H2O untuk *machine learning*, dan penyimpanan data dilakukan dengan basis data MySQL.

5. Uji Coba dan Evaluasi

Uji coba dilakukan dengan mencoba sistem yang telah dibuat untuk mencari kesalahan program. Salah satu bentuk uji coba yang dilakukan adalah melakukan perhitungan manual terhadap *dataset*, dan membandingkan dengan hasil dalam sistem.

Kemudian dilakukan verifikasi untuk memastikan sistem yang dibuat sudah sesuai dengan kebutuhan. Verifikasi dilakukan dengan metode survei terhadap mahasiswa Universitas Surabaya sebagai pengguna aplikasi dan wawancara terhadap *academic advisor* dan dosen di Universitas Surabaya.

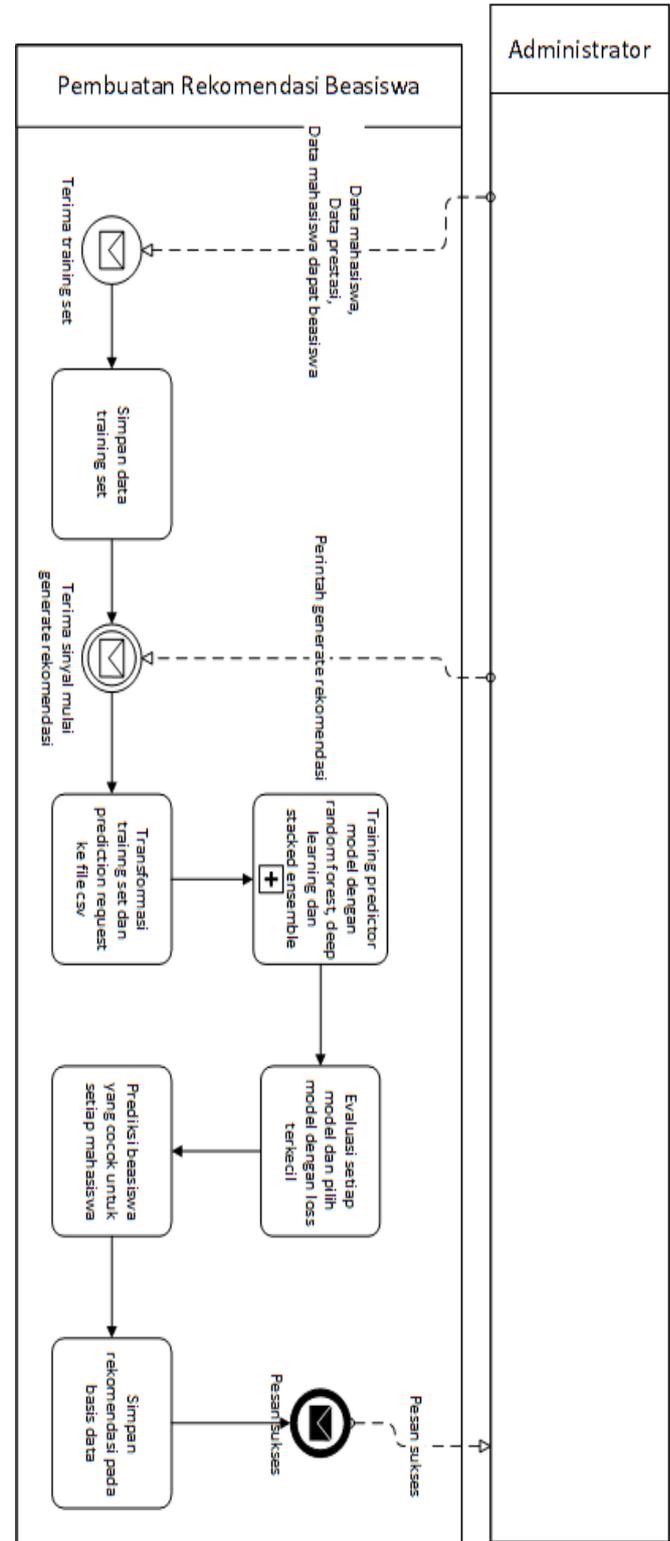
III. HASIL DAN PEMBAHASAN

Data set dalam penelitian ini diambil dari sistem terintegrasi yang ada di Universitas Surabaya, meliputi data penerima beasiswa, mahasiswa yang menjadi panitia kegiatan, mahasiswa yang menjadi peserta kegiatan, serta data akademis dan non akademis lainnya. Sedangkan desain data yang baru antara lain adalah struktur beasiswa, persyaratan beasiswa, dan kegiatan.

Fitur beasiswa termasuk menampilkan, mencari, melihat detail dan merekomendasikan beasiswa dengan *machine learning* dan metode rekomendasi manual dengan syarat, dimana desain proses dapat dilihat pada Gambar 1.

Algoritma *random forest* dipilih karena algoritma tersebut berdasar dari algoritma *decision tree*. Algoritma *decision tree* sendiri dipilih karena jumlah beasiswa yang ada pada Universitas Surabaya tidak banyak (dibawah 50 jenis beasiswa), sedangkan mahasiswa yang menerima dan yang harus diberi rekomendasi jauh lebih banyak. Sedangkan

algoritma *deep learning* dipilih karena akurasi tinggi untuk tugas klasifikasi. Algoritma *stacked ensemble* dipilih dengan tujuan menggabungkan hasil prediksi dari *random forest* dan *deep learning* untuk mencegah *overfitting* dan meningkatkan akurasi.



Gambar 1. Desain Proses Rekomendasi Beasiswa

Alasan pemilihan algoritma ini juga didasari bahwa data mahasiswa yang menerima beasiswa terlalu sedikit (*sparse*) sehingga algoritma sistem rekomendasi konvensional yang berbasis *implicit feedback* akan menghasilkan rekomendasi yang tidak presisi bagi mahasiswa yang belum pernah mendapat beasiswa. Di lain pihak, algoritma sistem rekomendasi konvensional yang berbasis *explicit feedback* seperti *collaborative filtering* membutuhkan setiap mahasiswa yang ingin mendapat rekomendasi yang presisi untuk sering memberi *feedback* berupa rating untuk setiap beasiswa. Selain itu algoritma rekomendasi konvensional yang berbasis *explicit feedback* juga kurang cocok diterapkan pada sistem yang memiliki banyak mahasiswa (*user*) tetapi memiliki sedikit beasiswa (konten).

Pada proses *training*, aplikasi menggunakan *automated machine learning* dengan *framework* H2O. Proses ini dimulai dengan *training* 1 buah model prediksi dengan algoritma *random forest* yang menggunakan parameter *default*. Parameter yang digunakan yaitu jumlah *tree* sebanyak 50, kedalaman maksimal 20, jumlah *histogram split position* sebanyak 20, dan rasio sampel baris 0,632. Rasio sampel baris 0,632 tersebut dipilih karena jika N buah sampel diambil dengan pengembalian sebanyak S kali, maka rata-rata jumlah sampel unik pada setiap percobaannya kurang lebih sebesar $0,63212056 \times N$ [14]. Selanjutnya aplikasi melakukan *training* 1 buah model *extremely random forest*, yaitu *random forest* yang menggunakan *feature sampling* sebelum memilih *best split*.

Aplikasi melakukan *training* 1 buah model *deep learning* dengan parameter *default*. Parameter yang digunakan yaitu fungsi aktivasi *Rectified Linear Unit* (ReLU), Jumlah *hidden layer* adalah 3 yang masing-masing memiliki 10 neuron, jumlah *epoch* sebanyak 10, *learning rate* sebesar 0,005. Urutan sampel pada *training set* tidak diacak. *Target class* (jenis beasiswa yang direkomendasikan) dikodekan dengan *one-hot encoding*.

Pada tahap selanjutnya, aplikasi melakukan *training* dengan *grid search* menggunakan algoritma *deep learning* hingga aplikasi mencapai jumlah total model yang terbentuk sebanyak 20. Istilah *grid search* menunjuk pada pencarian model terbaik dengan mencoba berbagai parameter dengan *range* tertentu. Parameter yang dicari berdasarkan *range* adalah jumlah *epoch*, fungsi aktivasi, jumlah *hidden layer*, dan jumlah neuron pada setiap *hidden layer* tersebut.

Model-model yang telah tercipta sebelumnya disebut model dasar. Setelah terbentuk 20 model dasar, aplikasi melakukan *training model* baru dengan algoritma *stacked ensemble* yang menggabungkan 20 model dasar. Selanjutnya aplikasi melakukan *training model stacked ensemble* terhadap 6 atau kurang model dasar yang memiliki *evaluation metric* terbaik. Model *stacked ensemble* yang terbentuk dari semua model digunakan untuk mencapai akurasi tertinggi. Sedangkan model *stacked ensemble* yang hanya menggunakan beberapa model terbaik digunakan untuk optimisasi agar model dapat membuat prediksi dengan lebih cepat.

Setiap model tersebut (baik model dasar maupun model *ensemble*) dievaluasi dengan *evaluation metric* yaitu *mean per class error*. Cara perhitungan *mean per class error* adalah

dengan rata-rata kesalahan prediksi pada setiap *class*. Data yang digunakan untuk evaluasi adalah data *cross-validation* dengan jumlah *fold* 5. Nilai evaluasi akhir diambil dari rata-rata *mean per class error* kelima model *cross-validation* tersebut. Keanggotaan sampel pada *fold* dipilih dengan syarat *modulo*, yaitu sampel nomor 1, 6, 11, 16, dan seterusnya menjadi anggota *fold* pertama, sampel nomor 2, 7, 12, 17, dan seterusnya menjadi anggota *fold* kedua, dan seterusnya hingga 5 *fold*.

Penerapan rekomendasi beasiswa yang dilakukan pada saat selesai *training* ini dipilih karena data mahasiswa yang mendapatkan beasiswa hanya akan bertambah atau berubah pada saat awal semester saja. Dengan metode ini aplikasi juga menjadi lebih cepat dalam memproses permintaan mahasiswa karena tinggal mengambil hasil rekomendasi dari basis data tanpa perlu menghitungnya setiap kali mahasiswa meminta rekomendasi.

Model dengan hasil evaluasi terbaik kemudian digunakan untuk memprediksi beasiswa yang cocok bagi seluruh mahasiswa. Meski demikian, hanya rekomendasi yang memiliki nilai *confidence* lebih besar dari 80 persen saja yang akan ditampilkan pada mahasiswa.

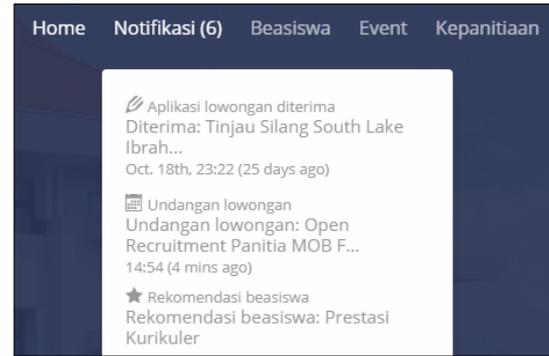
Fitur *event* termasuk menampilkan dan mencari *event* dalam bentuk daftar maupun bentuk kalender, melihat detail dan merekomendasikan *event* dengan metode rekomendasi manual. Fitur perekrutan menempel pada fitur aplikasi yaitu sebuah *event* dapat berupa *event* perekrutan, sehingga fitur-fitur *event* seperti menampilkan kalender, pencarian, dan rekomendasi dapat digunakan ulang untuk perekrutan. Fitur notifikasi terbagi menjadi notifikasi web yang akan ditampilkan pada saat mahasiswa masuk ke aplikasi web *student relationship management* dan notifikasi *Google Calendar* yang diintegrasikan dengan *G-Suite Gooya* milik akun mahasiswa. Pratinjau fitur beasiswa, *event* dan pencariannya, perekrutan dan notifikasi secara berurutan ditunjukkan pada gambar 2, gambar 3, dan gambar 4.

| Prestasi Kurikuler | |
|--------------------|---|
| Tujuan | Mendorong peningkatan prestasi akademik |
| Syarat Umum | Memperoleh IPS >= 3.0. Nilai tertinggi dari perhitungan IPS x konversi sks. Tidak berlaku untuk program internasional |
| Kuota | 1 orang per semester per prodi per angkatan |
| Besaran UPP | 100% |

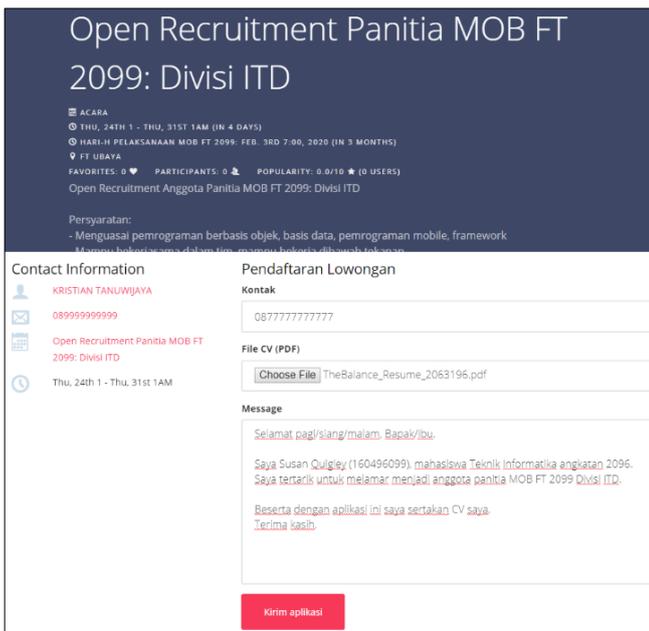
Gambar 2a. Pratinjau Fitur Beasiswa



Gambar 2b. Pratinjau Fitur Beasiswa



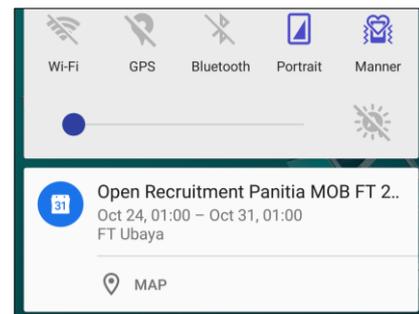
Gambar 4a. Pratinjau Fitur Notifikasi



Gambar 3a. Pratinjau Fitur Perekrutan



Gambar 3b. Pratinjau Fitur Perekrutan



Gambar 4b. Pratinjau Fitur Notifikasi

Uji coba sistem dilakukan melalui dua tahap, yaitu verifikasi dan validasi. Pada tahapan verifikasi, dilakukan uji coba terhadap 380 data mahasiswa penerima beasiswa, yang terbagi menjadi 64 penerima beasiswa simpati, 13 mahasiswa penerima beasiswa prestasi penalaran dan kreativitas, 71 mahasiswa penerima beasiswa olahraga dan seni, 86 mahasiswa penerima beasiswa prestasi kurikuler, dan 146 mahasiswa penerima beasiswa aktivis. Hasil uji coba menunjukkan rata-rata kesalahan setiap jenis beasiswa yang direkomendasikan adalah 0,3028 (30,28%) pada sampel yang belum pernah dilihat. Sedangkan rata-rata kesalahan dari *confusion matrix* menunjukkan rata-rata error sebesar 0,2368 (23,68%). Nilai ini berarti akurasi prediksi beasiswa terhadap data yang belum pernah dilihat adalah sebesar 76,32% atau dari 10 rekomendasi, lebih dari 7 rekomendasi sudah tepat.

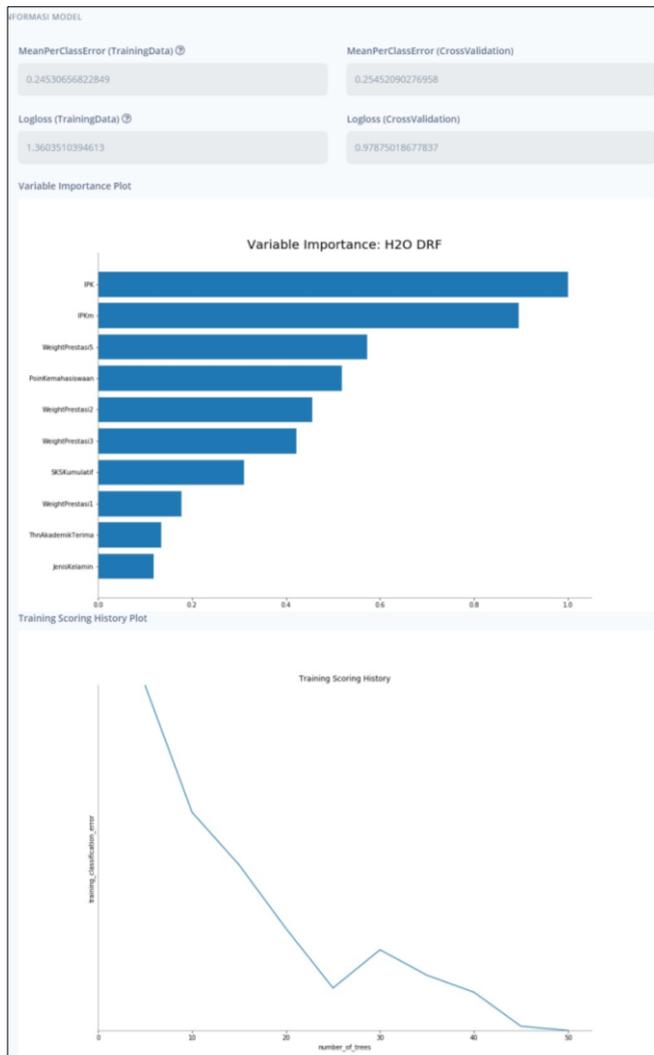
Pelaksanaan uji coba di atas adalah dengan melihat laporan hasil pembuatan rekomendasi sistem. Pengguna memilih menu 'SysRec Control Panel' pada halaman manajemen data lalu menekan tombol 'laporan machine learning' untuk menampilkan laporan tersebut. Aplikasi kemudian menampilkan halaman daftar model yang tercipta pada proses *automated machine learning* beserta hasil *scoring*.

Tabel tersebut diurutkan berdasarkan *metric* yaitu *mean per class error* terhadap data *cross-validation*, dimana model yang digunakan untuk membuat prediksi rekomendasi beasiswa adalah model teratas (terbaik). Nama model pada tabel tersebut menunjukkan algoritma model seperti DRF (*distributed random forest*), XRT (*extremely random forest*), *DeepLearning* dan *StackedEnsemble*.

Pada halaman detail model, aplikasi menampilkan informasi *scoring model* berupa *mean per class error* dan *logloss*, jenis algoritma yang digunakan untuk membuat model, tanggal

model dibuat, grafik kontribusi setiap *feature* (*variable importance plot*), dan grafik riwayat hasil *evaluation metric* setiap iterasi. Halaman tersebut juga menampilkan daftar parameter milik model tersebut. Jenis parameter yang ditampilkan bergantung pada jenis algoritma yang digunakan model tersebut, seperti model *random forest* akan memiliki parameter “*ntrees*” yang mengatur jumlah *decision tree* yang dibuat, sedangkan model *deep learning* akan memiliki parameter “*hidden*” yang mengatur jumlah neuron pada setiap *hidden layer*.

Tampilan halaman hasil uji coba menampilkan laporan *machine learning* beasiswa rekomendasi sistem berupa evaluasi model dan daftar parameter model ditunjukkan pada Gambar 5 dan Gambar 6. Tampilan uji coba tersebut menunjukkan model terbaik (*rank 0*) dari *automated machine learning* yang dilakukan. Algoritma yang digunakan model tersebut adalah *random forest* dengan parameter jumlah *decision tree* sebanyak 50, ketinggian *decision tree* tidak boleh lebih dari 20, dan rasio sampel yang digunakan untuk *training* setiap *decision tree* sebesar 0,632.



Gambar 5. Tampilan Laporan *Machine Learning* Rekomendasi Sistem

Pengujian dilakukan dengan data mahasiswa penerima beasiswa mulai dari tahun akademik 2015 gasal hingga tahun akademik 2018 genap. Jenis beasiswa pada data *training* adalah beasiswa prestasi kurikuler, beasiswa aktivis, beasiswa sosial ekonomi (*simpati*), beasiswa penalaran-kreativitas, dan beasiswa olahraga-seni. Hasil pengujian menunjukkan rata-rata kesalahan setiap beasiswa yang direkomendasikan adalah sekitar 0,2453 (atau 24,53%) yang berarti tiga dari empat beasiswa yang direkomendasikan sistem sudah tepat (akurasi 75,47%). Pengujian ini berlangsung selama 13396,267 detik (setara 3,7 jam) pada platform *Google Colaboratory Virtual Machine* yang memiliki spesifikasi RAM 12.72GB, *processor Intel Xeon x86 64-bit*, jumlah *core* 1 dengan *hyperthreading* (2 *logical core*), *clock speed* 2.30GHz.

Hasil uji coba juga menunjukkan grafik *variable importance plot* yang menunjukkan besaran kontribusi setiap *feature* terhadap prediksi rekomendasi beasiswa. Terlihat bahwa *variable* atau atribut yang berpengaruh paling besar adalah IPK dan IPK_m. Hal ini mampu dipahami secara intuitif karena beasiswa prestasi kurikuler sangat bergantung pada IPK, IPK_m, dan IPS dari calon penerimanya. Selanjutnya disusul dengan variabel prestasi 5 (kepanitiaan) dan jumlah poin kemahasiswaan yang juga dapat dipahami secara intuitif sebagai penentu penerima beasiswa aktivis.

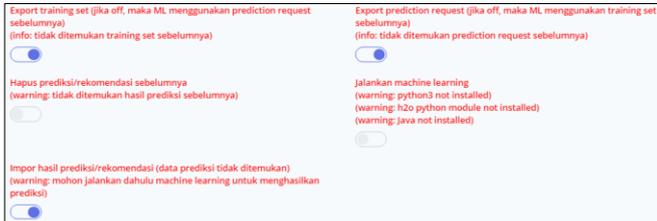
| PARAMETER | TYPE | NILAI | DEFAULT | LEVEL |
|---|----------|--------|---------|-----------|
| <i>n</i> fold | int | 5 | 0 | critical |
| <i>balance_classes</i> | boolean | FALSE | FALSE | secondary |
| <i>class_sampling_factors</i> | float[] | NULL | NULL | expert |
| <i>max_after_balance_size</i> | float | 5 | 5 | expert |
| <i>max_confusion_matrix_size</i> | int | 20 | 20 | secondary |
| <i>ntrees</i> | int | 50 | 50 | critical |
| <i>max_depth</i> | int | 20 | 20 | critical |
| <i>min_rows</i> | double | 1 | 1 | critical |
| <i>Row sample rate per tree (from 0.0 to 1.0)</i> | int | -1 | -1 | critical |
| <i>sample_rate</i> | double | 0.632 | 0.632 | critical |
| <i>sample_rate_per_class</i> | double[] | NULL | NULL | expert |
| <i>min_split_improvement</i> | double | 1.0E-5 | 1.0E-5 | secondary |

Gambar 6. Tampilan Laporan Parameter Model Rekomendasi Sistem

Uji coba selanjutnya adalah pembuatan rekomendasi sistem melalui *web interface*. Pengguna memilih menu ‘*SysRec Control Panel*’ pada halaman manajemen data. Aplikasi kemudian menampilkan *form* untuk menjalankan rekomendasi sistem. Pembuatan rekomendasi melalui *web interface* mampu menjalankan kelima tahap rekomendasi secara terpisah. Meski demikian, aplikasi juga menampilkan peringatan bahwa menjalankan rekomendasi sistem lebih disarankan untuk dilakukan dengan *command line interface* seperti yang telah disebutkan sebelumnya.

Pada proses pengaturan, aplikasi mendeteksi secara otomatis kebutuhan perangkat lunak untuk menjalankan rekomendasi sudah terpenuhi atau belum oleh sistem komputer yang menjadi HTTP *server* aplikasi. Jika terdapat tahap yang

kebutuhannya belum terpenuhi oleh server, aplikasi menampilkan pesan kesalahan berwarna merah pada nama tahap. Tampilan hasil uji coba pesan kebutuhan rekomendasi tidak terpenuhi oleh server ditunjukkan pada Gambar 7.



Gambar 7. Tampilan Pesan Kebutuhan Rekomendasi Tidak Terpenuhi

Uji coba dilanjutkan dengan menjalankan rekomendasi pada *web interface*. Aplikasi kemudian menjalankan perintah *console* yang sama dengan perintah menjalankan rekomendasi dengan *command line interface*, lalu menampilkan hasilnya pada halaman web. Hasil uji coba menjalankan rekomendasi melalui *web interface* ditampilkan pada Gambar 8.

```
Laporan STDOUT Hasil Running: Beasiswa Rekomendasi Sistem

>>> pwd
K:\Kristian\Documents\Ubaya\Kuliah\TA\project\public

>>> cd ..
.

>>> pwd
K:\Kristian\Documents\Ubaya\Kuliah\TA\project

>>> python3 storage/app/recommender/run_h2o_recommender_beasiswa.py

>>> php artisan python_export:beasiswa --prediction_request
Mengekspor prediction_request...
Berhasil mengekspor data csv

>>> php artisan python_clear:beasiswa
Berhasil menghapus beasiswa rekomendasi sistem

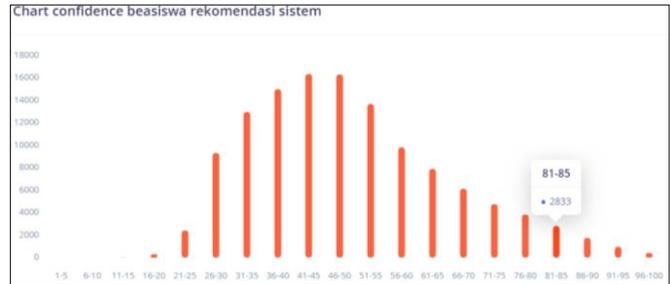
>>> python3 storage/app/recommender/run_h2o_recommender_beasiswa.py

>>> php artisan python_import:beasiswa
Berhasil mengimpor data rekomendasi
```

Gambar 8. Tampilan Hasil Menjalankan Rekomendasi Via Web

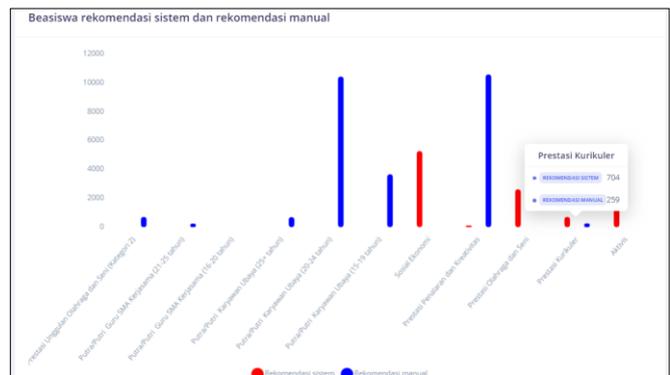
Setelah melakukan uji coba pembuatan beasiswa rekomendasi sistem, dilakukan uji coba visualisasi hasil rekomendasi tersebut dalam bentuk grafik. Grafik yang tersedia adalah *histogram* (grafik frekuensi) *confidence* dari rekomendasi dan *histogram* frekuensi jenis beasiswa yang

direkomendasikan. Angka frekuensi pada setiap grafik dapat dilihat dengan mengarahkan kursor pada salah satu batang. Pada *histogram* frekuensi jenis beasiswa, ditampilkan juga beasiswa rekomendasi manual. Tampilan hasil uji coba visualisasi data dalam bentuk *histogram confidence* dan *histogram* jenis beasiswa tersebut masing-masing ditampilkan pada Gambar 9 dan Gambar 10.



Gambar 9. Tampilan Grafik Confidence Beasiswa Rekomendasi Sistem

Pada tahap validasi, peneliti mengadakan survei pada 50 orang mahasiswa di Universitas Surabaya dari berbagai jurusan. Hasil survei menunjukkan bahwa 92% responden dapat menerima sistem ini karena dapat membantu mahasiswa dalam mendapatkan informasi beasiswa dan *event* yang *realtime* dan sesuai dengan kondisi mereka.



Gambar 10. Tampilan Grafik Jenis Beasiswa Rekomendasi Sistem

Hasil survei menunjukkan 100% responden mahasiswa setuju bahwa dengan adanya sistem rekomendasi beasiswa dan kegiatan ini, memudahkan mereka dalam melakukan pengajuan beasiswa maupun menentukan kegiatan yang akan diikuti. Pada uji coba terkait rekomendasi beasiswa, 76% responden mahasiswa menyatakan hasil rekomendasi sesuai dengan beasiswa yang pernah mereka dapatkan. Hal ini cukup sejalan dengan hasil uji coba verifikasi yang sebelumnya dilakukan. Uji subjektifitas terhadap perasaan responden dilakukan melalui kuisisioner, dengan mengajukan pertanyaan apakah sistem ini dapat meningkatkan perasaan lebih diperhatikan oleh institusi, 80% reponden menyatakan setuju.

IV. KESIMPULAN

Kesimpulan berikut dibuat dari hasil uji coba dan evaluasi sistem, serta analisis masalah. Dari dasar tersebut diambil kesimpulan bahwa konsep *Customer Relationship Management* dapat diimplementasikan ke sistem akademis, dengan tujuan untuk menjaga hubungan dengan mahasiswa. Selain itu, algoritma *machine learning* yaitu *random forest*, *deep learning* dan *stacked ensemble* dapat menjadi salah satu teknik yang diimplementasikan pada sistem CRM, yaitu dengan membantu memberikan rekomendasi hingga tingkat akurasi 75%.

Keterbatasan dari penelitian ini antara lain adalah belum dilakukannya dalam wilayah yang luas, uji coba dilakukan terbatas pada 380 *dataset* dan tingkat *error* 25%, sehingga akurasi masih dapat ditingkatkan dengan metode *machine learning* yang lebih sesuai.

REFERENSI

- [1] Soesiano, D. (2015). *Pembuatan Sistem Informasi Manajemen Hubungan Pelanggan pada Salon X Berbasis Web* [Tugas Akhir]. Surabaya, Jawa Timur: Perpustakaan Universitas Surabaya.
- [2] Tiono, W.J.H. (2015). *Pembuatan Sistem Informasi Manajemen Hubungan Pelanggan Apotek X Berbasis Web* [Tugas Akhir]. Surabaya, Jawa Timur: Perpustakaan Universitas Surabaya.
- [3] Winarta, E.F. (2017). *Customer Relationship Management pada PhotoStudio X Berbasis Website* [Tugas Akhir]. Surabaya, Jawa Timur: Perpustakaan Universitas Surabaya.
- [4] Seeman, E.D. & O'Hara, M. (2006). Customer Relationship Management In Higher Education: Using Information Systems To Improve The Student-School Relationship. *Campus-Wide Information Systems*, 23(1), pp. 24-34. DOI:10.1108/10650740610639714.
- [5] Wali, A.F. & Wright, L.T. (2016). Customer Relationship Management And Service Quality: Influences In Higher Education. *Journal of Customer Behaviour*, Vol. 15(1), pp. 67-79, <http://dx.doi.org/10.1362/147539216X14594362873532>
- [6] Rigo, G.E., Pedron, D.C., Caldeira, M. & Silva de Araújo, C.C. (2016). CRM Adoption in a Higher Education Institution. *JISTEM: Journal of Information Systems and Technology Management*, Vol. 13(1), pp. 45-60. DOI: 10.4301/S1807-17752016000100003
- [7] Buttle, F. (2009). *Customer Relationship Management: Concept and Technologies*. Burlington, Massachusetts: Elsevier Ltd.
- [8] Ali, J., Khan, R., Ahmad, N. & Masqood, I. (2012). *Random Forest and Decision Trees*. Diakses dari: <https://pdfs.semanticscholar.org/959a/8e906ee26b940374b719253c8e188ed78fd3.pdf>
- [9] Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks*, Vol. 61(1), pp. 85-117. DOI:10.1016/j.neunet.2014.09.003
- [10] Pernía-Espinoza A., Fernandez-Ceniceros, J., Antonanzas, J., Urraca, R. & Martinez-de-Pison, F.J., (2018). Stacking Ensemble With Parsimonious Base Models To Improve Generalization Capability In The Characterization Of Steel Bolted Components. *Applied Soft Computing*, Vol 70(1), pp. 737-750. DOI: 10.1016/j.asoc.2018.06.005.
- [11] Patni, S. (2017). *Pro RESTful APIs: Design, Build and Integrate with REST, JSON, XML and JAX-RS*. Santa Clara, California: Apress Media
- [12] Ricci, F., Rokach, L., Shapira, B. & Kantor, P. (2011). *Recommender System Handbook*. New York City, New York: Springer.
- [13] Grus, J. (2015). *Data Science from Scratch - First Principles with Python*. Sebastopol, California: O'Reilly Media.
- [14] Efron, B. & Tibshirani, R. (1997). Improvements on Cross-Validation: The 632+ Bootstrap Method. *Journal of the American Statistical Association*, Vol. 92(1), pp. 548-560. DOI: 10.1080/01621459.1997.10474007.