

Sistem Pembelajaran Pemrograman Memanfaatkan Konsep Skill Tree

Hanson Prihantoro Putro
Program Studi Teknik Informatika
Universitas Islam Indonesia
Sleman, Indonesia
hanson@uii.ac.id

Nur Muhammad
Alumni Program Studi Teknik Informatika
Universitas Islam Indonesia
Sleman, Indonesia
13523258@alumni.uui.ac.id

Abstract—Sudah banyak mahasiswa Teknik Informatika yang mengikuti kompetisi pemrograman dan belajar dari berbagai media dan sistem pembelajaran pemrograman yang tersedia secara daring. Namun demikian, banyak pula yang kurang bersemangat karena banyaknya materi algoritma dan bidang pemrograman yang harus dipelajari. Mahasiswa yang baru saja belajar pemrograman kesulitan untuk bisa mengimbangi rekannya yang telah belajar sejak sekolah menengah. Oleh karena itu, perlu strategi untuk membagi materi pembelajaran sebuah tim agar setiap anggotanya dapat fokus mempelajari satu jalur/bidang pemrograman. Terdapat sebuah teknologi yang dapat dimanfaatkan untuk menunjukkan keterhubungan antar materi yaitu dengan teknologi *skill tree*. Pada penelitian ini dikembangkan sebuah sistem pembelajaran pemrograman memanfaatkan konsep *skill tree* yang akan menampilkan susunan materi dalam bentuk pohon kemampuan. Sehingga untuk mencapai kemampuan tertentu, pengguna dapat mengerjakan materi dengan lebih runut dan terstruktur. Hasil yang didapatkan yaitu berupa perangkat lunak yang dapat digunakan untuk memodelkan peta konsep pemrograman dalam struktur pohon dan membantu proses pembelajaran pemrograman secara lebih runut dan terstruktur. Perangkat tersebut memodelkan *skill tree* ke dalam tiga bagian yaitu: 1) *Node* berupa konsep pembelajaran; 2) *Edge* berupa keterhubungan antar konsep; dan 3) *Activity* yaitu kegiatan belajarnya seperti membaca, mengerjakan kuis dan juga soal pemrograman.

Kata kunci—sistem pembelajaran; pembelajaran pemrograman; *skill tree*; LMS.

I. PENDAHULUAN

Sebagai sarana untuk saling berkompetisi, program studi Teknik Informatika di berbagai perguruan tinggi di Indonesia telah berpartisipasi dalam berbagai kontes dan lomba pemrograman (*competitive programming*). Namun demikian, mayoritas peringkat teratas yang menjuarai kompetisi tersebut masih diisi oleh mahasiswa yang berasal dari minoritas perguruan tinggi [1]. Para mahasiswa yang menjuarai kompetisi telah mempelajari materi pemrograman sejak mereka mengikuti olimpiade komputer untuk siswa sekolah menengah [2]. Mereka kemudian masuk ke perguruan tinggi favorit mereka. Sedangkan perguruan tinggi yang kurang menjadi favorit tidak mendapatkan input mahasiswa yang baik terkait keilmuan pemrograman ini. Mengingat pentingnya berada dalam lingkungan kompetisi pemrograman [3], para

dosen Teknik Informatika perlu merespon perbaikan pembelajaran pemrograman untuk mahasiswanya, baik dari segi materi maupun metode pembelajaran.

Terdapat banyak algoritma yang harus dipahami untuk mengikuti kompetisi pemrograman agar mendapatkan hasil yang maksimal [4]. Secara praktek, algoritma-algoritma tersebut dapat dibagi kepada tiap anggota tim untuk dipelajari secara terpisah. Hal ini mempermudah para mahasiswa yang baru saja mempelajari pemrograman untuk fokus pada beberapa algoritma atau bidang pemrograman saja untuk mengejar ketertinggalan. Saat mengikuti kompetisi pemrograman, mereka pun dapat berbagi soal sesuai dengan bidang yang mereka pelajari.

Di sisi lain, media pembelajaran yang ada belum mengakomodasi pengguna untuk fokus pada bidang pemrograman tertentu. Sistem pembelajaran yang ada khususnya di Indonesia [5] menyusun materinya secara linear. Sistem ini mengharapkan pengguna untuk mempelajari dan mengerjakan semua materi pemrograman secara berurutan, tanpa ada hubungannya dengan fokus bidang yang ditargetkan. Hal ini menyulitkan mahasiswa dalam belajar algoritma pemrograman untuk keperluan kompetisi. Banyaknya algoritma yang harus dipelajari mengurangi semangat mahasiswa untuk belajar dan juga membuat pembelajaran menjadi kurang terarah.

Pembelajaran pemrograman dapat lebih terarah pada fokus bidang tertentu apabila mahasiswa mengetahui keterhubungan antar materi pemrograman. Sebuah materi pemrograman dapat menjadi prasyarat sebelum mempelajari materi yang lain. Dengan satu materi menjadi prasyarat dari materi-materi yang lain, dapat disusun sebuah struktur pohon yang memperlihatkan kebergantungan antar materi. Konsep pohon yang merepresentasikan keterhubungan antar materi populer digunakan dalam sistem permainan [6], disebut sebagai pohon teknologi (*technology tree*) atau lebih umumnya pohon kemampuan (*skill tree*). Dalam sistem pembelajaran, konsep ini masih jarang digunakan. Salah satu dari yang ada tersebut menggunakan struktur pohon untuk mempelajari bahasa asing [7].

Selama ini mahasiswa belajar pemrograman melalui berbagai sistem manajemen pembelajaran LMS (*Learning Management System/LMS*), baik yang disediakan secara umum dalam skala internasional, nasional, maupun yang

disediakan oleh perguruan tinggi khusus untuk mahasiswanya. Pemanfaatan LMS secara kolaboratif dalam membantu proses pembelajaran dan memberikan kesuksesan yang lebih baik dibandingkan pembelajaran tradisional [8]. Namun demikian belum ada LMS yang menggunakan konsep *skill tree* untuk menyajikan materi pembelajarannya. Oleh karena itu perlu dikaji, bagaimana mengimplementasikan konsep *skill tree* pada sistem manajemen pembelajaran untuk membantu pembelajaran pemrograman.

Penelitian ini bertujuan untuk mengembangkan sebuah alat bantu pembelajaran pemrograman yang menggunakan konsep *skill tree*. Selanjutnya makalah ini membahas rancangan sistem pembelajaran yang memanfaatkan *skill tree* serta penggunaan dari LMS tersebut. Diharapkan konsep ini dapat memperkaya LMS pemrograman yang sudah ada sehingga dapat memudahkan mahasiswa untuk belajar pemrograman dengan lebih terarah.

II. PEKERJAAN TERKAIT

Learning management system adalah infrastruktur yang mengelola konten instruksional dan menilai pembelajaran individu dan organisasi serta melacak kemajuan dalam memenuhi suatu tujuan hingga menyajikan data untuk mengawasi proses pembelajaran pada sebuah organisasi. Karakteristik umum dari sebuah LMS dalam pendidikan yaitu tujuan instruksional yang terikat pada pelajaran individu, pelajaran-pelajaran yang dimasukkan ke dalam kurikulum standar, sistem manajemen pengumpulan hasil dan performa mahasiswa, materi tambahan dalam beberapa tingkatan serta pelajaran-pelajaran yang disediakan berdasarkan proses pembelajaran setiap mahasiswa [9].

Berbagai penelitian terkait pengembangan LMS telah dilakukan, termasuk di antaranya yang kemudian digunakan khusus untuk pembelajaran pemrograman. Sebuah sistem pembelajaran telah dikembangkan untuk pelatihan kompetisi pemrograman [10]. Sistem ini kemudian menjadi pondasi yang digunakan untuk pembelajaran pemrograman khususnya untuk kompetisi pemrograman serta olimpiade komputer dalam skala nasional [5]. Penelitian terkait LMS pemrograman ini dilakukan pula pada pengembangan arsitektural [11] dan skalabilitas sistem [12].

Penelitian lain terkait pengembangan LMS untuk belajar pemrograman masih terbatas pada pengembangan arsitektural sistem [11] [13] serta pengembangan fitur dan materi ajar [14] [15]. Sebuah kajian telah memetakan berbagai LMS untuk belajar pemrograman dengan beberapa kategori pengembangan yaitu: 1) mesin kompilasi daring (*online compiler/autograder*); 2) data mining; 3) pembelajaran teknologi informasi secara umum 4) penilaian dalam kompetisi pemrograman 3) penilaian rekrutmen pegawai; 4) kerangka pengembangan sistem (*development platforms*) [16].

Belum ada penelitian terkait metode belajar dalam pengembangan LMS pemrograman, terlebih menggunakan konsep *skill tree*. Hanya terdapat satu sistem pembelajaran pemrograman daring yang menggunakan struktur pohon sederhana [17], namun masih dengan materi dan keterhubungan yang terbatas. Sistem tersebut belum menggali keterhubungan antar materi dengan maksimal.

Skill tree adalah elemen permainan yang terdiri dari pohon bertingkat (*precedence tree*) di mana setiap *node* pohon mewakili tugas tertentu yang akan memberikan poin pengalaman kepada pemain apabila telah menyelesaikannya. Penerapan elemen permainan dapat memberikan efek positif dalam proses pembelajaran [18]. *Skill tree* menjadi salah satu konsep yang dikaji dalam pembelajaran multimedia, memberikan sumbangan performa 40% lebih baik dibandingkan yang tidak. Bidang lain yang telah menerapkan *skill tree* adalah pembelajaran bahasa asing di sebuah aplikasi pembelajaran daring populer [7]. Penelitian ini mengkaji pengembangan sistem pembelajaran pemrograman dengan memanfaatkan konsep *skill tree*.

III. METODOLOGI

Untuk mengembangkan LMS sehingga dapat memanfaatkan *skill tree* dalam menyajikan materi pembelajarannya maka yang pertama dilakukan adalah mengkaji *skill tree* seperti apa yang akan diimplementasikan. Oleh karena itu, perlu dirancang *skill tree* yang mengadopsi konsep-konsep yang telah digunakan pula dalam berbagai sistem permainan dan disajikan dalam literatur yang telah dikaji.

Beberapa hal yang harus disiapkan untuk merancang sebuah *skill tree* yaitu:

- Daftar *node* pohon atau kemampuan (*skill*) yang disediakan.
- Hirarki atau keterhubungan prasyarat (*link*) antar kemampuan yang ada.
- Permainan atau aktivitas yang harus dilakukan dalam setiap *node* kemampuan agar pemain (*user*) dianggap telah menguasai kemampuan tersebut.

Kebutuhan terkait *skill tree* ini digali dari staf pengajar perguruan tinggi, baik yang mengajarkan pemrograman di kelas maupun yang sudah berpengalaman di dunia kompetisi pemrograman. Dari rancangan ini kemudian diimplementasikan ke dalam sistem pembelajaran pemrograman atau LMS yang menggantikan penyajian materi pemrograman dari struktur linier ke struktur pohon.

Metode penelitian dilanjutkan dengan pembagunan sistem pembelajaran pemrograman dengan model waterfall menggunakan pendekatan berorientasi objek mulai dari analisis, perancangan, implementasi hingga pengujian. Tahap perancangan dibantu dengan bahasa permodelan UML dan arsitektur MVC. Implementasi dilakukan dengan membuat sebuah prototipe aplikasi web sistem pembelajaran pemrograman yang dapat langsung diujicobakan. Kemudian aplikasi diujikan, termasuk kepada staf pengajar dan mahasiswa untuk mendapatkan berbagai kesan dan masukan untuk mengoptimalkan penggunaan sistem.

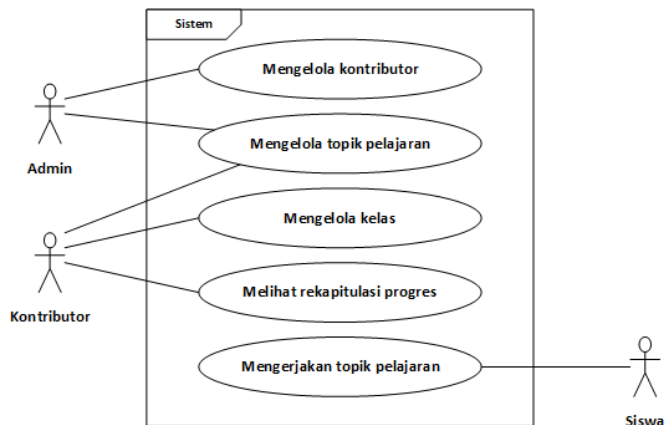
IV. PERANCANGAN SISTEM

A. Model Kebutuhan

Dalam mengembangkan sistem pembelajaran, diidentifikasi peran yang ada di dalam sistem yang dibangun. Pada proses identifikasi, diperoleh empat peran yaitu: siswa

atau mahasiswa, kontributor dan admin. Siswa adalah pengguna yang akan mengerjakan topik-topik pelajaran. Sedangkan kontributor adalah pengguna yang diperankan oleh staf pengajar yang akan mengelola kelas, mengisi topik pelajaran dan melihat rekapitulasi progress siswa. Dan admin adalah pengguna sebagai spesialisasi dari staf pengajar yang akan mengelola kontributor dan topik pelajaran.

Setelah diidentifikasi aktor yang berperan, kemudian dirancang *use case* yang memodelkan kebutuhan perangkat lunak yang telah dibuat sebelumnya. Dirancang empat *use case* yaitu mengelola kontributor, mengelola topik pelajaran, mengelola kelas, melihat rekapitulasi progres dan mengerjakan topik pelajaran seperti diperlihatkan pada Gambar 1.



Gambar 1. Diagram *use case* sistem pembelajaran pemrograman memanfaatkan *skill tree*

Mengelola kontributor dilakukan oleh admin yang berupa menambah, mengubah dan menghapus data kontributor. Admin dan kontributor dapat mengelola topik pelajaran, yaitu admin dapat menambah, mengubah, menghapus, mengisi detail topik pelajaran dan mengkonstruksi pohon kemampuan, sedangkan kontributor hanya dapat mengisi detail pelajaran. Kontributor juga dapat mengelola kelas yang terdiri dari menambah, mengubah, menghapus dan memasukkan siswa ke dalam kelas tertentu. Selain itu, hal yang dapat dilakukan oleh kontributor yaitu melihat rekapitulasi progres yang terbagi menjadi dua bagian yaitu melihat rekapitulasi progres per kelas dan per siswa dalam bentuk pohon kemampuan. Adapun siswa dapat mengerjakan topik pelajaran yang terdiri dari mengerjakan topik pelajaran beserta melihat capaian kemampuan yang telah diraih dalam bentuk pohon kemampuan.

B. Rancangan Skill Tree

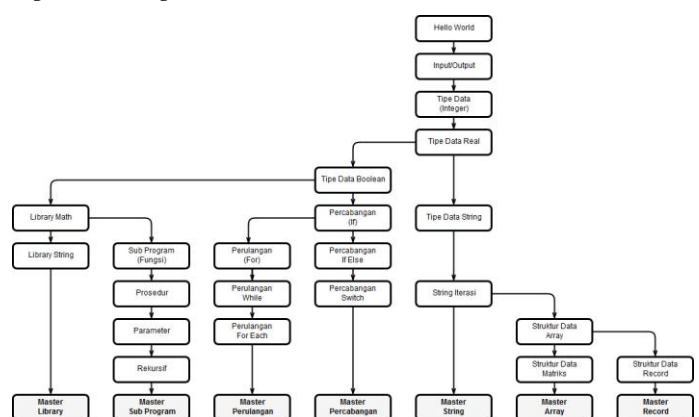
Skill tree menjadi objek utama sekaligus inovasi yang disertakan di dalam sistem pembelajaran yang akan dikembangkan. Oleh karena itu, pemanfaatan *skill tree* harus dirancang dengan baik. Dengan menggali kebutuhan mengenai *skill tree* dari literatur dan staf pengajar, diperoleh rancangan *skill tree* seperti diperlihatkan pada Tabel 1.

TABEL 1. PEMETAAN RANCANGAN KONSEP *SKILL TREE*

| No | Konsep Tree | Konsep Skill Tree | Rancangan Sistem |
|----|-----------------------|------------------------------------|--|
| 1 | Titik (<i>node</i>) | Kemampuan (<i>skill</i>) | Tema/topik/konsep pembelajaran, bidang pemrograman, algoritma, dst. |
| 2 | Garis (<i>egde</i>) | Keterhubungan (<i>link</i>) | Prasyarat bahwa suatu topik yang terhubung di atasnya harus dikuasai sebelum dapat mempelajari topik di bawahnya |
| 3 | - | Aktivitas (<i>game/activity</i>) | Materi, kuis, soal pemrograman, dst. |

Secara umum, sebuah struktur data pohon akan memiliki titik dan garis. Titik dalam sistem ini disepakati sebagai konsep pembelajaran seperti kemampuan melakukan pembacaan dan penulisan ke standard input/ouput, tipe data, percabangan, perulangan, prosedur, fungsi dan lain sebagainya. Beberapa konsep tersebut dapat dibagi kembali menjadi beberapa *node* agar lebih fokus dalam pembelajarannya, misalnya percabangan yang memiliki tiga skema, maka akan dirancang tiga *node* untuk percabangan.

Garis sebagai *direct edge* yang selalu mengarah ke bawah menunjukkan keterurutan proses pembelajaran dari atas ke bawah. Jika sebuah kemampuan belum selesai dipelajari dan dikuasai, maka kita tidak akan dapat mempelajari kemampuan di bawahnya. Hal ini mirip dengan sistem yang memiliki konsep keterurutan dalam pembelajarannya. Namun yang membedakan adalah bahwa saat satu kemampuan selesai dikuasai, bisa jadi terdapat beberapa kemampuan lain di bawahnya. Pengguna dapat memilih salah satu dari kemampuan tersebut dan mengabaikan kemampuan yang lain jika memiliki keterbatasan waktu dalam pembelajaran. Termasuk jika pengguna juga sudah memiliki target ingin menguasai (*mastering*) salah satu konsep pembelajaran saja, pengguna akan mengetahui alur pembelajarannya (*path*). Contoh keterhubungan titik dan garis untuk pembelajaran Pemrograman Dasar atau materi Algoritma dan Pemrograman diperlihatkan pada Gambar 2.



Gambar 2. Contoh hasil rancangan *skill tree* untuk pembelajaran Pemrograman Dasar

Sebagai sebuah pohon kemampuan, konsep ini memiliki aktivitas yang bisa dipilih di dalam titik tersebut. Dengan demikian, sistem dirancang agar pohon ini menjadi sebuah pohon yang dinamis. Pengguna dapat mengklik *node* yang

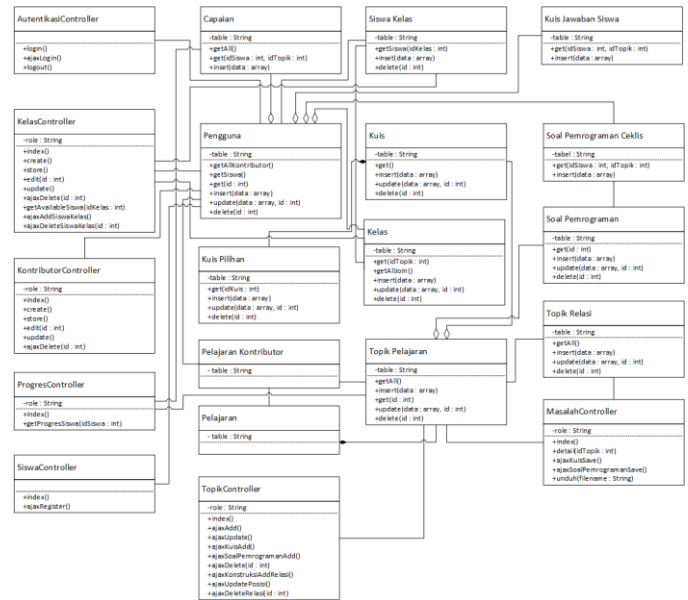
tersedia (*available*) dari *skill tree* tersebut. *Node* yang tersedia adalah yang *node* di atasnya juga sudah tersedia. Oleh karena itu, *node* paling atas pertama kali harus tersedia. Setelah diklik, pengguna dapat memilih aktivitas yang dapat diselesaikan agar ia dapat membuat *node* di bawahnya juga tersedia. Dari penggalan kebutuhan sebelumnya, terdapat tiga aktivitas yang harus dikerjakan: 1) Membaca materi terkait konsep pembelajaran tersebut; 2) Mengerjakan soal kuis yang berupa pilihan ganda; 3) Mengerjakan soal pemrograman di mana pengguna harus membuat program (*coding*). Setelah ketiga pekerjaan tersebut selesai dilakukan, maka

Sistem yang akan dibangun adalah LMS yang sudah ada dengan materi pembelajaran/kemampuan disusun secara hirarkis dalam struktur pohon, bukan linear. Akan tetapi terdapat kebutuhan bahwa karena untuk mencapai suatu materi/kemampuan pemrograman memungkinkan memiliki persyaratan lebih dari satu materi/kemampuan lainnya. Oleh karena itu digunakan struktur pohon yang mengizinkan satu *node* mempunyai banyak induk. Secara istilah struktur data, struktur ini merupakan struktur *Graph*, bukan *Tree*. Namun dalam konsep pembelajaran, struktur ini masih disebut sebagai pohon (*Tree*).

C. Struktur Kelas

Setiap *use case* tersebut kemudian dirancang kelas-kelas yang bersesuaian. Pertama yaitu Mengelola Kontributor, tentunya akan ada entitas kontributor yang tidak lain adalah Pengguna juga. Kemudian Mengelola Topik Pelajaran, di dalam inilah terdapat *skill tree*. Dengan demikian sesuai rancangan maka akan perlu Pelajaran, Topik dan Relasi. Selanjutnya Mengelola Kelas, di dalamnya terdapat Kelas itu sendiri, beserta Pengguna sebagai siswa atau bisa juga mahasiswa. Lalu Melihat Rekapitulasi Progres, sistem harus memiliki Capaian pembelajaran untuk setiap siswa. *Use case* terakhir yaitu mengerjakan topik pembelajaran. Maka di dalamnya akan terdapat aktivitas yang harus diselesaikan seperti Materi, Kuis dan Soal Pemrograman.

Dari identifikasi tersebut kemudian dirancang berbagai kelas yang dibutuhkan sistem hingga mendapatkan sebuah kelas diagram (Gambar 3). Struktur kelas ini dirancang menggunakan arsitektur MVC sehingga dari entitas yang ada kemudian dilengkapi dengan *Model*, *View* dan *Controller*. Dalam diagram kelas *view* tidak disertakan karena banyaknya *view* yang ada. Kelas *model* akan berperan untuk melakukan *query* ke basis data serta memiliki atribut yang eksplisit. Kesemuanya dikendalikan oleh kelas *controller*.



Gambar 3. Rancangan kelas sistem pembelajaran pemrograman memanfaatkan *skill tree*

V. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Seluruh dokumentasi perancangan kemudian diimplementasikan sehingga menghasilkan perangkat lunak aplikasi web yang siap dijalankan. Implementasi perangkat lunak menggunakan bahasa pemrograman PHP dengan bantuan *framework* CodeIgniter. Implementasi dari *use case* memungkinkan untuk mengelola topik pelajaran memungkinkan melakukan penambahan, pengubahan, mengisi detail, menghapus dan mengkonstruksi pohon kemampuan. Dari penggunaan tersebut, terdapat beberapa hal yang perlu diperhatikan sebuah sistem pembelajaran yang menggunakan *skill tree*.

1. Membangun *Skill Tree* (Gambar 4)

a. Menambahkan topik ke pohon

Pengguna atau guru/staf pengajar/pengelola dapat menambahkan sebuah topik pembelajaran baru ke *skill tree* yang sudah ada. Secara *default*, topik ini akan berada di posisi paling bawah pohon.

b. Mengatur posisi topik dalam tampilan

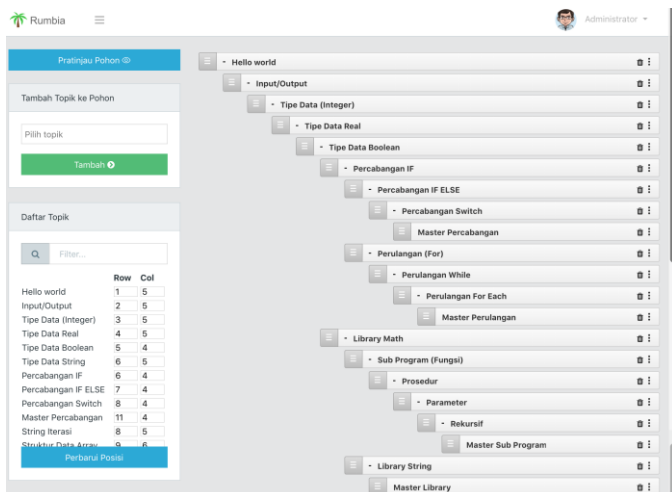
Dari *libray* yang diperoleh, posisi dari topik tidak bisa diatur secara otomatis. Oleh karena itu, perlu diatur secara manual dengan menentukan posisi vertikal/level dengan posisi *row* dan menentukan posisi horizontal dengan posisi *col*.

c. Mengatur keterhubungan antar topik

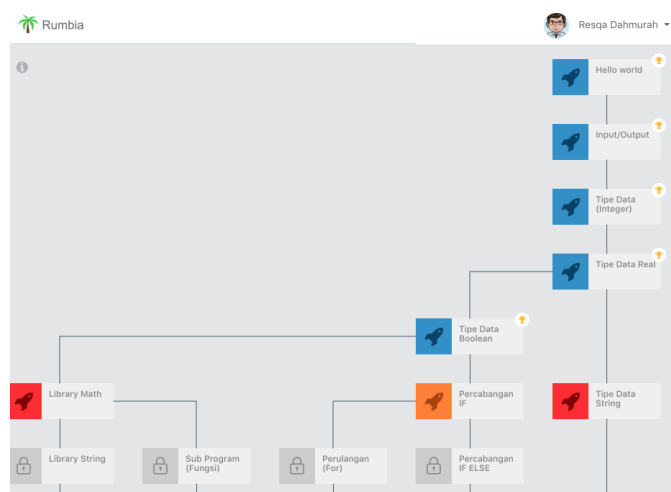
Keterhubungan antar topik dapat diatur dengan menyeret (*drag*) suatu topik menjadi berada di bawah topik lain yang akan menjadi induknya.

d. Mengelola isi dari topik.

Setelah topik dibuat, pengguna juga dapat mengelola isi dari topik tersebut, berupa materi, kuis dan soal pemrograman.



Gambar 4. Implementasi konstruksi pohon kemampuan



Gambar 5. Implementasi daftar topik dan capaian

2. Mengerjakan topik dalam *skill tree* (Gambar 5)
Pengguna dalam hal ini siswa dapat mengerjakan *node* atau topik yang tersedia. Setelah sebuah topik dibuka, terdapat tiga bagian topik yaitu materi yang dapat dibaca, kuis yang terdiri dari beberapa soal pilihan ganda serta soal pemrograman di mana siswa diminta untuk membuat program dari kasus yang diberikan.
3. Menentukan tingkatan pencapaian setiap topik
Sebagai salah satu aspek dalam permainan juga yaitu nilai. Dalam sistem ini, diberikan pula nilai bagi pengguna yang telah melakukan aktivitas dalam sebuah topik serta tingkatan atau levelnya. Warna merah menunjukkan bahwa topik belum dikerjakan (*not-attempt*), warna kuning menunjukkan tingkat pemula, warna orange menunjukkan tingkat menengah, warna hijau menunjukkan tingkat mahir dan warna biru menunjukkan tingkat ahli. Setiap tingkatan menandakan prosentasi aktivitas yang telah dikerjakan. Tingkatan ahli berarti semua aktivitas sudah dikerjakan.
4. Menentukan aturan *passing grade* aktivitas untuk ke topik berikutnya
Seperti disampaikan bahwa sebuah topik tidak dapat dibuka jika topik di atasnya belum diselesaikan. Namun demikian jika harus menunggu semua aktivitas di atasnya selesai, hal ini bisa mengurangi semangat pengguna karena beberapa soal dirancang hanya sebagai tambahan. Oleh karena itu pada sistem ini, topik di bawahnya akan tersedia jika pengguna telah dalam tingkatan mahir dengan prosentasi kemajuan pengerjaan dalam prosentase tertentu.
5. Menguasai konsep secara terfokus
Dengan struktur pohon tersebut, maka pengguna dapat memilih ujung konsep yang akan dikuasai. Setelah target ditentukan maka ia dapat mempelajari hanya konsep-konsep yang berada dalam jalur mulai dari topik pembelajaran pertama hingga topik terakhir tersebut. Hal ini tentunya akan membantu para mahasiswa yang ingin segera mengejar ketertinggalan dalam menguasai sebuah algoritma untuk bisa mengikuti kompetisi pemrograman. Sedangkan rekan mahasiswanya yang lain dapat mempelajari topik pelajaran pada jalur yang berbeda yang akan menguntungkan bagi tim mereka.

B. Pengujian Sistem

Setelah diimplementasikan, selanjutnya dilakukan pengujian. Pengujian terdiri dari dua pengujian yaitu pengujian berdasarkan use case dan terhadap pengguna. Pengujian berdasarkan use case dilakukan berdasarkan kelima use case, untuk selanjutnya dari use case tersebut diperoleh rangkaian kasus uji yang siap untuk diuji. Pada tahap ini dapat diketahui pemenuhan kebutuhan sistem yang dibuat apakah sudah berjalan dengan baik atau tidak. Kesimpulan yang didapatkan yaitu dari total 95 kasus uji, 95 kasus uji berhasil dan tidak ada kasus uji yang gagal. Dengan demikian, persentase keberhasilan yang diperoleh yaitu 100%.

Adapun pengujian terhadap pengguna dilakukan kepada staf pengajar dan mahasiswa yang dibedakan berdasarkan perannya masing-masing terhadap sistem. Pengujian terhadap staf pengajar dilakukan kepada dua orang staf pengajar, sedangkan untuk mahasiswa dilakukan kepada 20 orang mahasiswa program studi Teknik informatika yang tersebar ke dalam beberapa tingkat yaitu 6 orang pada tahun pertama, 4 orang tahun kedua, 4 orang tahun ketiga dan 6 orang tahun keempat. Pengujian dilakukan dengan meminta pengguna untuk menggunakan sistem untuk selanjutnya memberikan saran dan komentar.

Hasil pengujian terhadap staf pengajar adalah staf pengajar lebih mudah untuk mengatur keterhubungan antar materi pembelajaran dan lebih mudah dalam memantau perkembangan progress mahasiswanya dalam mengerjakan topik yang berikan. Para staf juga merasa bahwa dengan *skill tree* ini, peta konsep pembelajaran dari mata kuliah apapun dapat dimodelkan dan dibuatkan *skill tree*-nya di sistem tersebut.

Sedangkan bagi mahasiswa, hasil pengujian yang diperoleh adalah bahwa sistem ini dapat membantu untuk belajar dengan lebih runut dan terstruktur dengan adanya pohon kemampuan yang diberikan. Mereka juga lebih antusias dalam mengerjakan aktivitas-aktivitas yang ada. Belajar konsep yang diharapkan menjadi lebih cepat. Namun demikian, masih terdapat kekurangan minor pada sistem. Tampilan dan desain interaksi dengan pengguna masih perlu diperbaiki.

VI. KESIMPULAN

Telah berhasil dibangun sebuah sistem manajemen pembelajaran yang memanfaatkan konsep *skill tree*. Sistem yang dibangun berhasil memodelkan peta konsep pemrograman dalam struktur pohon. Sistem ini dapat membantu proses pembelajaran pemrograman menjadi lebih runut dan terstruktur. Dengan desain yang sudah dirancang untuk LMS dengan konsep *skill tree*, sistem memungkinkan pengelola untuk mengatur keterhubungan setiap topik pelajaran ke dalam bentuk pohon kemampuan. Siswa juga dapat terbantu dalam mempelajari konsep yang ada, bersemangat berkat pengarahan yang diberikan oleh *skill tree*.

Beberapa pekerjaan ke depan yang akan dilakukan yaitu melakukan pengujian untuk mengetahui sejauh mana efektivitas pembelajaran materi-materi kompetisi pemrograman yang disusun menggunakan *skill tree* oleh sistem ini. Kemudian akan dilakukan pula perbaikan sistem seperti penambahan fitur *autograder* pada soal pemrograman agar tidak terjadi kecurangan saat mengerjakan soal pemrograman. Selain itu, akan dilakukan juga perbaikan desain UI/UX agar sistem lebih mudah digunakan.

REFERENSI

- [1] Binus University, "Scoreboard ACM-ICPC Indonesia National Contest 2017", *Competition Binus University*, 2017. [Online]. Available: <https://competition.binus.ac.id/inc2017/>. [Accessed: 06- Jul- 2018].
- [2] Tim Olimpiade Komputer Indonesia, "About » Tim Olimpiade Komputer Indonesia", *Toki.id*, 2011. [Online]. Available: <https://toki.id/about/>. [Accessed: 06- Jul- 2018].
- [3] O. Astrachan, "Non-competitive programming contest problems as the basis for just-in-time teaching", *Frontiers in Education*, vol. 34, pp. T3H-20, 2004.
- [4] S. Halim and F. Halim, *Competitive programming*, 3rd ed. Singapore: Lulu Independent Publish, 2013.
- [5] Tim Olimpiade Komputer Indonesia, "Course 1: Pemrograman Dasar", *TLX Training Gate*, 2018. [Online]. Available: <https://training.ia-toki.org/training/curriculums/1/courses/1/chapters>. [Accessed: 06- Jul- 2018].
- [6] T. Ghys, "Technology trees: Freedom and determinism in historical strategy games", *Journal of Computer Game Research*, vol. 12, no. 1, 2012.
- [7] I. Garcia, "Learning a language for free while translating the web. Does duolingo work?", *International Journal of English Linguistics*, vol. 3, no. 1, 2013.
- [8] N. Cavus, "Assessing the success rate using a learning management system together with collaborative tool in web based teaching programming languages", *Educational Computing Research* vol. 36, no. 3, pp. 301-321, 2007.
- [9] W. R. Watson and S. L. Watson, "An argument for clarity: what are learning management systems, what are they not, and what should they become?" *TechTrends Springer Verlag*, vol. 51, no. 2, pp 28-34, 2007.
- [10] P. Novandi, "Sistem Pelatihan Kompetisi Pemrograman Komputer", Undergraduate, Institut Teknologi Bandung, 2009.
- [11] J. Fernando and I. Liem, "Components and Architectural Design of an Autograder System Family", *Olympiads in Informatics*, vol. 8, pp. 69-79, 2014.
- [12] K. Danutama and I. Liem, "Scalable autograder and LMS integration", *Procedia Technology*, vol. 11, pp. 388-395, 2013.
- [13] D. Milojicic, "Autograding in the cloud: Interview with David O'Hallaron", *IEEE Internet Computing*, vol. 15, no. 1, pp. 9-12, 2011.
- [14] A. N. Pradana, J. Adisantoso, "Pengembangan sistem manajemen pembelajaran pemrograman bahasa Scheme, Java, PHP, dan Perl", *Jurnal Ilmu Komputer dan Agri-Informatika*, vol. 2, no. 2, pp 47-54, 2013.
- [15] M. Ayub, M. C. Wijayanto and W. F. Senjaya, "Edukasi berpikir komputasional melalui pelatihan guru dan tantangan Bebras untuk siswa di Bandung pada tahun 2016", *Prosiding Seminar Nasional Pengabdian Kepada Masyarakat*, vol. 2, no. 2, 2017.
- [16] S. Wasik et al., "A survey on online judge systems and their applications", *ACM Computing Surveys*, vol. 51, no. 1, pp. 1-34, 2018
- [17] CodeFight, "Practice Programming and Find Tech Jobs - CodeFights", *Codefights.com*, 2018. [Online]. Available: <https://codefights.com/arcade>. [Accessed: 07- Jul- 2018].
- [18] G. Barata et al., "Relating gaming habits with student performance in a gamified learning experience", *ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play*, Toronto, Canada, pp. 17-25, 2018.