

TRIZ: TRIMMING APPROACH PADA SOFTWARE PROSES IMPROVEMENT

Kholid Haryono¹

¹ Jurusan Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia,
Jl. Kaliurang KM. 14.4 Sleman Yogyakarta 55584. telp (0274) 898444; fax (0274) 898459

¹kholid.haryono@uii.ac.id

Abstrak: Pandemi Covid-19 telah memaksa seluruh institusi berbenah dan berubah karena adanya pembatasan temu fisik di semua prosesnya. Proses yang memungkinkan tidak temu fisik adalah meningkatkan kualitas dan proses yang dilakukan *software (Software Process Improvement)* untuk menjalankan proses bisnisnya. TRIZ merupakan perangkat *problem solving* yang memiliki banyak *tool* seperti *Trimming*, *Function Analysis*, *Severity Level* dan *Impact* yang dapat digunakan untuk mencari peluang inovasi terhadap *software*. Kasus yang digunakan adalah salah satu fitur sistem akademik yang memiliki banyak *stakeholder* dan berdampak besar di masa pandemi sehingga dituntut untuk melakukan peningkatan (*improvement*). Hasil implementasi telah menemukan tiga proses yang dapat ditingkatkan. Peningkatannya terbagi menjadi dua yaitu melakukan *trimming* (memangkas) proses berjumlah dua, dan meningkatkan kualitas berjumlah satu proses. Peningkatan ini membuktikan bahwa TRIZ berhasil digunakan dalam proses SPI untuk *improve* sistem guna meningkatkan proses di dalam organisasi.

Kata Kunci: SPI, TRIZ, *Trimming*, *Function Analysis*, *Severity Level*, Sistem Akademik.

Abstract: The Covid-19 pandemic has forced all institutions to clean up and change due to restrictions on physical gatherings in all processes. The process that allows no physical meeting is to improve the quality and processes carried out by *software (Software Process Improvement)* to run its business processes. TRIZ is a problem solving tool that has many tools such as *Trimming*, *Function Analysis*, *Severity Level* and *Impact* that can be used to find innovation opportunities for *software*. The case used is one of the features of the academic system that has many stakeholders and has a big impact during the pandemic, so it is demanded to be improved. Implementation results can improve processes by trimming the previous two processes and improving the quality of one other main process.
Keywords: SPI, TRIZ, *Trimming*, *Function Analysis*, *Severity Level* and *Impact*, System of Academic

A. PENDAHULUAN

Pandemi memaksa kegiatan dan aktivitas dalam organisasi dilakukan secara *online* untuk menghindari pertemuan fisik. Hal ini menuntut pengelola IT pada semua institusi untuk

meningkatkan kualitas layanan IT di masa krisis[1], terutama *software* yang menjalankan proses bisnis agar menghindari pertemuan fisik dan atau kerumunan yang membahayakan. Diperlukan kegiatan peningkatan *software* yang digunakan agar dapat menjamin berjalannya proses bisnis organisasi selama pandemi[2].

Peningkatan proses bisnis melalui *software* dapat dilakukan dalam dua hal yakni peningkatan proses dan peningkatan kualitas *software*[3]. Bagaimana *improvement* dapat dilakukan dengan baik perlu pendekatan dan tahapan pada suatu kasus[4]. Pemilihan pendekatan dan tahapan dapat mempengaruhi kualitas peningkatan proses dan kualitas *software*[3].

Penelitian ini mencoba melakukan praktik pendekatan dan tahapan pada lingkungan akademik kampus, khususnya proses yang melibatkan *stakeholder* yang banyak dan

membutuhkan temu fisik. Proses tersebut adalah pengambilan Rencana Akademik Semester (RAS). Proses ini melibatkan mahasiswa, bagian akademik, bank tempat membayar, dan Dosen Pembimbing Akademik (DPA).

Proses ini terjadi pada setiap awal semester. Dimulai dari perubahan status hak mengambil semester baru tersebut, mahasiswa melakukan pembayaran di bank yang telah bekerjasama dengan kampus dan disediakan loket kasir di kampus, mahasiswa melakukan penyusunan RAS, menghadap DPA, proses persetujuan DPA dan RAS terbit menjadi jadwal mata kuliah per mahasiswa per semester. Proses yang cukup panjang dan mengeluarkan usaha lebih (sibuk) untuk semua yang terlibat. Oleh karenanya kasus ini sesuai menjadi studi kasus dalam usaha peningkatan *software*.

Penelitian ini akan menjawab masalah bagaimana meningkatkan kualitas proses dan sistem akademik agar di masa pandemi ini semua proses bisnis dapat berjalan dengan lebih baik. Tujuannya membuat sebuah model proses yang lebih cepat dan prosesnya pendek serta mencari peluang untuk menemukan titik proses yang dapat ditingkatkan kualitasnya.

B. LANDASAN TEORI

Penelitian ini didukung oleh beberapa teori yang akan memperkuat hasil penelitian. Teori tersebut meliputi: SPI, TRIZ, *Trimming*, *Function Analysis*, dan *Severity Level and Impact*.

A. *Software Process Improvement (SPI)*

Bidang SPI diketahui sejak tahun 1930 melalui hasil kerja dari ahli statistik kontrol, Shewhart. Hasil pekerjaan ini ditemukan oleh Deming dan Juran yang mengatakan bahwa proses yang berkualitas itu merupakan inti dari pengiriman

produk yang berkualitas. Mereka percaya bahwa produk yang berkualitas merupakan hasil dari proses yang berkualitas. Selanjutnya, hasil dari Deming dan Juran ini diimplementasikan pada bidang kualitas *software (Software Quality)* oleh Watts Humphrey dan lainnya di Software Engineering Institute (SEI). Ini yang menjadi titik awal kelahiran SPI [3].

SPI sendiri didefinisikan sebagai sebuah program dari aktivitas-aktivitas yang didesain untuk meningkatkan kinerja dan kedewasaan proses-proses dalam organisasi melalui penggunaan *software* dan hasil dari program tersebut [3]. Fokusnya mendukung organisasi dalam mencapai tujuan bisnis yang lebih efektif, di-*deliver* ke pasar lebih cepat, peningkatan kualitas yang terus menerus, dan mengurangi atau mengeliminasi hal-hal yang tidak berguna.

Pencapaian tujuan organisasi dapat dilakukan dalam dua hal yakni meningkatkan (*improvement*) atau mengurangi (*reduction*). Peningkatan dapat berupa meningkatkan kualitas, produktivitas, ketepatan pengiriman, konsistensi biaya dan jadwal, kepuasan pelanggan, sikap baik karyawan dan sebagainya. Sedangkan pengurangan dapat berupa biaya atas kualitas yang jelek, biaya pengembangan *software*, dan pengurangan lain yang merugikan.

Framework yang digunakan dapat disesuaikan dengan lingkungan pengembangan. SEI menawarkan lima tahap mulai dari perencanaan, implementasi, *piloting*, *deploy*, dan identifikasi peningkatan selanjutnya[3]. Petersen menggunakan SPI-LEAM (*Lean Measurement*) dengan enam tahap yaitu: memahami *current system*, tentukan tujuan dan pengukuran, memilih proses dan model yang akan digunakan, eksekusi, analisis hasil, dan membungkus (*packaging*) dan

pengiriman[5]. Galinac membuat strategi implementasi SPI ke dalam enam tahapan yaitu: mengidentifikasi peluang (*opportunity*), isi (*content*), pendekatan (*approach*), eksekusi, validasi, dan integrasi[6]. Tahap tersebut disarikan dari dua pendekatan yakni IDEAL[7] dan QIP[8]. Pada penelitian ini, tahap yang digunakan adalah enam tahap dari Galinac karena sesuai dengan lingkungan studi kasus yang diambil pada penelitian ini.

Banyak model yang dapat digunakan di antaranya adalah *Capability Maturity Model Integration* (CMMI), Standar ISO 9001, ISO 15504 (SPICE), *Root Cause Analysis* (RCA), *Six sigma*, *Balanced score card*[3]. Belum banyak ditemukan SPI menggunakan pendekatan TRIZ. Penelitian ini akan mempraktikkan metode ini sebagai perangkat yang *powerful* dan banyak digunakan untuk proses-proses inovasi produk.

B. TRIZ

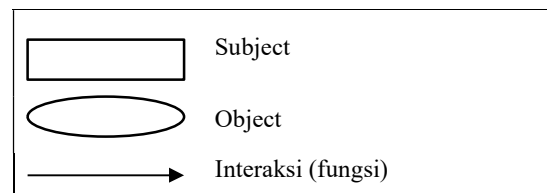
TRIZ merupakan akronim dari bahasa Rusia yakni *Teoriya Resheniya Izobreatatelskikh Zadatch*. Dalam bahasa Inggris dikenal dengan *Theory of Inventive Problem Solving* (TIPS). Adalah suatu metode penyelesaian masalah yang terstruktur dan sistematis berbasis logis dan data, bukan intuisi atau aktivitas yang bersifat spontanitas[9][10][11]. Awalnya TRIZ diimplementasikan pada lingkungan industri atau manufaktur[12][13], selanjutnya berkembang ke bidang lain seperti bidang layanan, manajemen, dan bisnis[9]. TRIZ juga digunakan di banyak proyek IT untuk menyelesaikan berbagai persoalan proses yang terjadi di dalam proyek[14].

C. Function Analysis (FA)

Salah satu perangkat TRIZ yang berguna untuk melakukan analisis fungsi antara dua atau lebih komponen di dalam sistem[9]. Suatu sistem

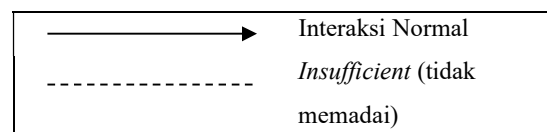
bekerja menghasilkan *output* yang diharapkan karena adanya interaksi fungsi antar komponen yang ada di dalamnya. Untuk memahami fungsi setiap komponen maka perlu dianalisis fungsi interaksi pada masing-masing komponen. Secara prinsip, fungsi antar komponen merupakan hubungan antara subject dan object. *Subject* adalah pihak pelaku yang menjalankan fungsi sedangkan *object* adalah pihak yang dilayani. Antara *subject* dan *object* terdapat garis interaksi yang menggambarkan fungsi subject melakukan sesuatu untuk *object*. Oleh karenanya, fungsi dibuat dalam bentuk kata kerja (*verb*). Misalnya tangan memegang *mouse*. Tangan adalah *subject*, dan *mouse* adalah *object* sedangkan “memegang” adalah fungsinya.

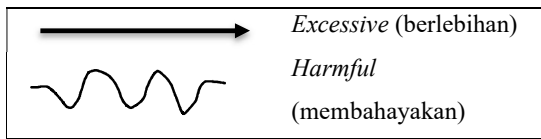
Terdapat tiga simbol penting pada analisis fungsi sebagaimana ditunjukkan pada Gambar 1. Simbol kotak persegi panjang menunjukkan komponen pelaku atau subyek. Komponen ini bertugas menjalankan suatu fungsi atau aksi terhadap obyek. Simbol kedua oval adalah pihak yang dilayani (obyek). Sedangkan garis panah mendeskripsikan suatu fungsi seperti memegang, membersihkan, dan fungsi lainnya.



Gambar 1. Simbol utama FA

Terdapat empat tipe garis interaksi pada FA. Masing-masing menunjukkan efek atau dampak interaksi. Simbol garis ditunjukkan pada Gambar 2.





Gambar 2. Simbol garis fungsi pada FA

Pada Gambar 2, tiga garis interaksi yang menjadi perhatian pada proses *improvement* yaitu garis *Insufficient*, *Excessive*, dan *Harmful*. Penyelesaiannya dapat berupa *improvement* maupun *reduction* (pengurangan/eliminasi) [15][9][11].

D. Trimming Approach

Merupakan salah satu perangkat TRIZ yang *powerful* dapat membantu mengeliminasi komponen yang tidak menguntungkan. *Trimming* dapat meningkatkan efisiensi dan bahkan mengurangi biaya proses[12][9].

Terdapat tiga tipe aturan pada *trimming* yaitu: Aturan A dimana object sudah tidak membutuhkan lagi fungsi tersebut. Misalnya Compact Disk (CD) yang saat ini tidak ada peminatnya, maka CD Player dapat di-*trimming* karena sudah tidak ada lagi pelanggan yang membutuhkan.

Aturan B dimana obyek atau pengguna fungsi dapat melakukan sendiri fungsi tersebut (*self service*). Misalnya *teller* bank yang melayani transfer, pengambilan atau penyetoran uang. Fungsi transfer dana sudah dapat dilakukan sendiri oleh nasabah dari mesin ATM. Bahkan ATM juga tidak lagi berfungsi karena melalui *mobile banking*, nasabah dapat melakukan sendiri berbagai layanan perbankannya.

Aturan C di mana suatu fungsi ditransfer ke komponen lainnya sehingga komponen sebelumnya dapat di-*trimming*. Misalnya *mouse* dapat di-*trimming* karena fungsi *mouse* atau *mouse pad* telah ditransfer ke layar dengan fitur *touchscreen*.

E. Severity Level and Impact

TRIZ mengenalkan pengelompokan tingkat keparahan (*Severity Level*) dan dampak merugikan yang ditimbulkan dari suatu fungsi yang ditunjukkan melalui garis interaksi. Tingkat keparahan pada setiap interaksi antar komponen sistem tersebut dapat dikelompokkan menjadi empat tingkatan. Masing-masing tingkatan menggunakan nilai kelipatan 5 dan dimulai dengan angka 5. Tingkatan tersebut ditunjukkan pada Tabel 1.

Tabel 1. Severity Level dan Impact

Tingkat Keparahannya (<i>Severity level</i>)	Value	Impact
Efek ringan (<i>slight effect</i>)	5	Efek ringan dari sistem atau performa luaran sistem.
Efek signifikan (<i>significant effect</i>)	10	Efek signifikan dari sistem atau performa luaran sistem. Subsystem bisa jadi berdampak.
Efek mayor (<i>major effect</i>)	15	Semua kinerja sistem mengalami penurunan kinerja meskipun masih bisa dioperasikan. Diperlukan rencana aksi dan perbaikan.
Efek berbahaya (<i>Hazardous effect</i>)	20	Sistem mati, <i>down</i> , dan tidak dapat beroperasi.

Tingkatan pada Tabel 1 akan digunakan untuk menilai interaksi antar komponen pada analisis fungsi. Tingkat keparahan yang diukur adalah tiga interaksi yang dinilai bermasalah yakni interaksi yang *Insufficient*, *Excessive*, dan *Harmful*.

C. METODOLOGI PENELITIAN

Model dan tahapan penelitian ini mengikuti enam fase yang digunakan oleh Tihana Galinac. Enam tahapan tersebut meliputi: tahap *opportunity*, tahap *content*, tahap *approach*, tahap *execution*, tahap *validation*, dan tahap *integration*. Pada penelitian ini, tahap integrasi tidak dilakukan karena baru sampai usulan desain proses.

A. Tahap Opportunity

Tahap ini digunakan untuk mengidentifikasi peluang-peluang untuk melakukan *improvement* terhadap *software* yang digunakan. Proses identifikasi dilakukan dengan mendeskripsikan studi kasus yang akan digunakan yaitu proses mahasiswa melakukan pengambilan mata kuliah pada setiap awal semester. Kegiatan ini disebut Keyin RAS (Rencana Akademik Semester). Kasus ini diambil dari praktik yang dilakukan oleh sebuah perguruan tinggi swasta di Indonesia.

Proses keyin RAS melibatkan empat pihak yakni mahasiswa sebagai pelaku utama yang akan mengambil mata kuliah pada semester tersebut; petugas akademik dan keuangan yang mengaktifkan semester dan membuat tagihan pembayaran; petugas bank yang melayani pembayaran; dan dosen pembimbing akademik (DPA) yang memeriksa rencana studi dan memberikan persetujuan. Hasil persetujuan DPA sekaligus sebagai finalisasi (*submit*) RAS mahasiswa.

Pada masa normal sebelum pandemi Covid-19, beberapa proses masih membutuhkan temu fisik seperti pembayaran di bank, mahasiswa mengisi RAS bersama mahasiswa lain, dan bimbingan dengan DPA. Pada masa pandemi, aktivitas fisik tersebut harus dibatasi bahkan dieliminasi. Di samping itu, proses RAS yang melalui beberapa tahapan berpotensi dapat dipangkas agar proses dapat berjalan lebih efektif.

B. Tahap Content

Tahap ini digunakan untuk menentukan tujuan dan ruang lingkup peningkatan yang diharapkan. Berdasarkan uraian pada tahap sebelumnya didapatkan bahwa beberapa proses yang telah berjalan harus diubah menyesuaikan kondisi pandemi (kualitas proses menjadi lebih efektif).

Pada proses perubahan tersebut, juga berkesempatan untuk meningkatkan proses yang sudah ada menjadi lebih baik (kualitas *software*). Sehingga, tujuan yang ditetapkan dalam proses *improvement* pada penelitian ini ada dua yakni meningkatkan proses dan meningkatkan kualitas *software*.

Ruang lingkup proses fokus terhadap proses mana saja yang akan dieliminasi sehingga hasilnya lebih pendek dan membutuhkan waktu lebih cepat. Sedangkan kualitas *software* lebih fokus pada proses mana yang bisa ditingkatkan (tidak dieliminasi) sehingga menjadi lebih baik.

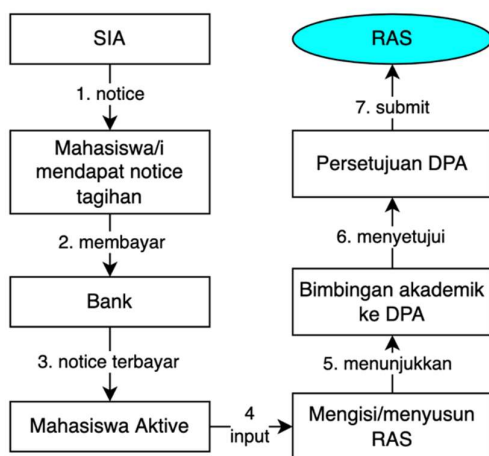
C. Tahap Approach

Terdapat beberapa metode dan perangkat yang digunakan dalam melaksanakan proses *improvement* tersebut. Pertama, menjabarkan proses RAS yang telah berjalan (*current system*) untuk mengetahui konteks sistem yang akan di-*improve* menggunakan TRIZ dengan metode analisis fungsi (*Function Analysis - FA*). Kedua, menggunakan *Trimming Approach* untuk mengusulkan ide perubahan yang akan dilakukan. Ketiga, menggunakan pengukuran *severity level* pada analisis bagian sistem yang akan dilakukan *improvement*.

Pada tahap ini fokus memahami sistem untuk proses RAS menggunakan FA. Sedangkan metode lainnya digunakan pada tahap *Execution* dan *Validation*. Untuk memahami proses RAS, perangkat TRIZ yang digunakan adalah analisis fungsi (*function analysis*). Terdapat tiga tahapan utama dalam melakukan analisis fungsi. Pertama menganalisis komponen apa saja yang terdapat dalam keseluruhan proses sistem. Kedua menganalisis fungsi interaksi antar komponen di dalam sistem. Ketiga, menggambarkan model

fungsi yang lengkap sehingga terlihat secara komprehensif keseluruhan fungsi di dalam sistem.

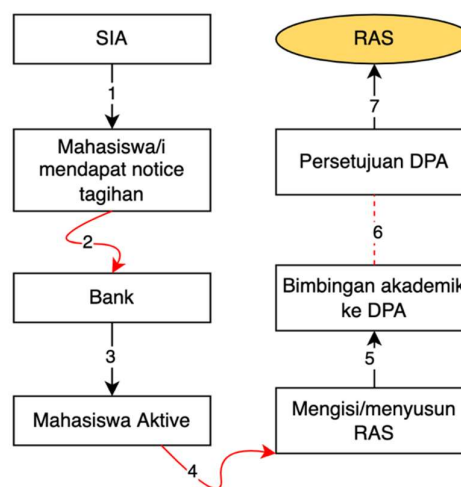
Hasil dari identifikasi komponen di dalam sistem ditunjukkan pada Gambar 3.



Gambar 3. Analisis Fungsi sistem RAS

Gambar 3 terdiri dari tujuh komponen subyek, tujuh garis fungsi, dan satu produk atau obyek. Untuk memahami proses pada *software*, masing-masing komponen dihubungkan menggunakan garis panah yang menunjukkan fungsi seperti komponen SIA mengirimkan *notice* ke mahasiswa terkait kewajiban membayar. Garis mahasiswa ke bank mewakili fungsi membayar, begitu seterusnya.

Langkah selanjutnya melakukan analisis interaksi (*Interaction Analysis*) untuk mengidentifikasi peluang titik yang akan ditingkatkan. Hasil analisis interaksi antar komponen ditunjukkan pada Gambar 4.



Gambar 4. Analisis interaksi

Setiap awal semester, user admin Sistem Akademik (SIA) mengaktifkan periode RAS dan meng-*generate* tagihan mahasiswa.

D. Tahap Execution

Penerapan *Trimming* untuk meningkatkan proses dan kualitas *software* dilakukan pada tahap ini. Gambar 4 menunjukkan bahwa dari tujuh interaksi, terdapat tiga interaksi yang perlu ditingkatkan yaitu interaksi nomor 2 dan nomor 4 dengan sifat *harmful*, dan interaksi nomor 6 yang bersifat *insufficient*. Ketiga interaksi tersebut dinilai level keparahannya dengan memberikan skor *severity level* oleh praktisi TRIZ. Skor keparahan ditunjukkan pada Tabel 2.

Tabel 2. Severity Level interaksi fungsi

Fungsi	Value	Sifat interaksi	Tipe Trimming
2. Mahasiswa membayar ke bank	15	Harmful	Tipe C
4. Mahasiswa mengisi/menyusun RAS	10	Harmful	Tipe A
6. DPA memeriksa dan menyetujui RAS	10	Insufficient	Tipe B

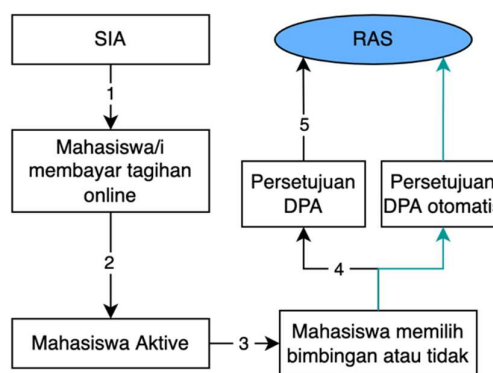
Fungsi 2 mahasiswa membayar di bank dapat membahayakan, terutama layanan bank di luar

tempat tinggal. Bertemunya secara fisik dengan berbagai pihak baik fasilitas umum, petugas bank, maupun sesama nasabah menjadi sulit dihindarkan. Asesmen praktisi TRIZ menyatakan bahwa *trimming* Tipe C dapat dilakukan yaitu fungsi ke bank untuk membayar tidak diperlukan karena telah ditransfer ke perangkat gawai yang dimiliki oleh mahasiswa melalui layanan pembayaran *mobile*. *Channel* pembayaran perbankan juga telah menyediakan banyak kemudahan dan berbasis *mobile banking*.

Fungsi 4 mahasiswa mengisi atau menyusun RAS yang biasanya dilakukan bersama mahasiswa lain atau dosen pembimbingnya dilakukan secara fisik, sehingga dapat membahayakan satu dengan lainnya. *Trimming* tipe A direkomendasikan yaitu fungsi tidak diperlukan karena sistem dapat menyusun otomatis berdasarkan kriteria setiap mata kuliah per semester dan mencakup keseluruhan yang dipersyaratkan dalam kurikulum. Ini disebut sistem rekomendasi otomatis pemilihan mata kuliah per semester.

Sedangkan Fungsi 6 yang merupakan proses bimbingan dan persetujuan pada kenyataanya hanya diperlukan untuk mahasiswa yang bermasalah dengan studinya dan membutuhkan konsultasi. Bagi mahasiswa yang prestasi akademiknya sudah bagus dan sesuai dengan alur kurikulum yang diikutinya maka proses bimbingan itu hanya pilihan dan tidak wajib. Sehingga, bisa memilih persetujuan otomatis. Oleh karenanya, fungsi 6 bersifat *insufficient*. *Trimming* tipe B dapat dipilih karena sebagian fungsi DPA dapat dilakukan secara mandiri oleh mahasiswa.

Setelah proses *trimming* dilakukan maka model fungsi yang diusulkan ditunjukkan pada Gambar 5.



Gambar 5. Model fungsi yang diusulkan

E. Tahap Validation

Validasi dilakukan melalui penilaian dua praktisi TRIZ yang terlibat pada proses sistem akademik. Pada Gambar 5 yang merupakan hasil akhir dari model fungsi setelah dilakukan *trimming* menunjukkan bahwa terdapat dua fungsi yang di-*trimming* dan satu fungsi yang ditingkatkan. Proses pembayaran yang tidak lagi melibatkan datang langsung ke bank dapat dilakukan melalui implementasi *Host to Host* dengan bank baik melalui kebijakan *virtual account* maupun integrasi antar layanan menggunakan API kedua sistem, sistem tagihan kampus dan pembayaran perbankan.

Fungsi otomasi pengisian RAS dapat dilakukan dengan implementasi sistem rekomendasi pengambilan mata kuliah semester yang dapat dibuat berdasarkan capaian lulusan kurikulum yang berlaku. Penelitian terkait model rekomendasi akan menjadi peluang penelitian selanjutnya dengan memperhatikan karakteristik kurikulum dari berbagai prodi termasuk prodi dari luar perguruan tinggi pada studi kasus ini.

Fungsi bimbingan dan persetujuan dapat dijadikan pilihan sehingga prosesnya semakin pendek karena hanya mahasiswa bermasalah saja yang membutuhkan bertemu dengan DPA.

D. HASIL DAN PEMBAHASAN

A. Hasil

Setelah tahapan penelitian dilaksanakan dan dijabarkan pada metodologi penelitian, maka dihasilkan dua yang utama yakni perubahan proses RAS dan peluang kualitas *software*. Pada perubahan proses RAS dari sebelumnya tujuh tahap menjadi lima tahap. Terdapat dua tahap yang di-*trimming* untuk meningkatkan proses yakni tahap dua dan tahap empat seperti ditunjukkan pada Gambar 4 dan 5. Selain itu, terdapat satu fungsi yang ditingkatkan melalui *trimming* tipe B yakni fungsi 6 yaitu DPA memeriksa dan menyetujui RAS.

Peningkatan kualitas *software* dapat dilakukan terutama pada fungsi 4 yakni mengeliminasi proses pengisian atau penyusunan RAS oleh mahasiswa. Peningkatan dapat dilakukan melalui penerapan sistem rekomendasi otomatis. Di samping itu, fungsi 6 juga dapat diturunkan kompleksitas prosesnya dengan menjadikan proses bimbingan dan persetujuan DPA sebagai pilihan, bukan kewajiban.

Berdasarkan empat sifat interaksi pada proses analisis interaksi yakni normal; kurang (*insufficient*); bahaya (*harmful*); dan berlebihan (*excessive*). Maka, didapatkan empat interaksi normal, dua interaksi *harmful*, dan satu interaksi *insufficient*. Interaksi yang *harmful* langsung dilakukan *trimming* sedangkan interaksi *insufficient* dilakukan *trimming* sebagian. Hasil akhir model fungsi ditunjukkan pada Gambar 3.

B. Pembahasan

Peningkatan proses dilakukan melalui penerapan TRIZ yakni metode FA yang memiliki tiga tahap yakni menggambarkan fungsi seluruh komponen yang terlibat pada sistem; melakukan analisis interaksi antar komponen; dan membuat

model fungsi. Melalui FA ini maka sistem secara komprehensif dapat dipahami.

Hasil analisis interaksi yang ditunjukkan pada Tabel 2 dapat digunakan sebagai dasar dalam proses pemilihan tipe *trimming*. Tipe *trimming* dapat mempermudah praktisi saat memilih berbagai alternatif solusi yang sesuai dengan konteks komponen yang akan di-*trimming*. Skor level *severity* pada Tabel 2 didapat dari proses asesmen oleh dua praktisi TRIZ.

Peningkatan kualitas *software* pada *trimming* tipe A fungsi 4 bermakna bahwa proses pengisian dan penyusunan RAS tidak lagi dibutuhkan oleh mahasiswa karena dapat dilakukan melalui proses otomatis. Pola kurikulum yang dibagi menjadi per semester dengan plotting mata kuliah yang jelas pada setiap semester menjadi peluang otomatisasi tersebut. Bahkan sistem dapat menampilkan antarmuka keadaan kurikulum yang telah ditempuh mahasiswa untuk setiap semesternya.

Jadi mahasiswa diberikan tampilan kontrol mata kuliah dari semester satu hingga delapan, mana saja yang sudah diambil, nilainya apa, sudah lulus atau belum, apa saja SKP yang pernah didapat, aktivitas wajib apa saja yang sudah dan belum diambil seperti KKN, pelatihan internal prodi dan lainnya. Tampilan juga memberikan prediksi tercepat kelulusan yang dapat ditempuh berdasarkan informasi kontrol mata kuliah semester tersebut. Kesalahan pemilihan mata kuliah langsung berdampak pada prediksi lamanya waktu studi.

Terkait implementasi TRIZ pada SPI dapat dilakukan secara lebih baik karena berangkat dari analisis fungsi setiap komponen yang terlibat pada setiap proses. Fokus terhadap fungsi pada setiap tahapan SPI dapat mempercepat proses *Improvement*.

E. KESIMPULAN

Berdasarkan hasil analisis fungsi dan implementasi TRIZ menggunakan FA dan *Trimming*, maka didapatkan hasil peningkatan proses yang cukup signifikan dengan memotong dua proses dari tujuh menjadi lima. Kualitas software dapat ditingkatkan dengan implementasi sistem rekomendasi pengisian atau penyusunan RAS secara otomatis berdasarkan konsiderasi yang diatur pada setiap kurikulum. Sistem rekomendasi ini menjadi peluang penelitian selanjutnya. Selain berhasil mengeliminasi dua proses, melalui FA juga telah berhasil meningkatkan fungsi keenam yakni proses monitoring dan persetujuan.

Beberapa saran untuk penelitian berikutnya adalah peluang mengembangkan sistem prediksi dan rekomendasi pengambilan mata kuliah secara otomatis. Sistem juga berpeluang memberikan prediksi kelulusan dari setiap mata kuliah yang diambil dan telah diselesaikan. Selain itu, pengembangan sistem bimbingan yang memberikan profil mahasiswa bagi pembimbing juga dapat diotomasi dalam penelitian lebih lanjut.

UCAPAN TERIMA KASIH

Terimakasih untuk seluruh TIM yang ada pada Badan Sistem Informasi UII atas semua bantuan informasi dan asesmen selama proses penelitian ini.

REFERENSI

- [1] S. V. Zykov, "IT crisisology: The new discipline for managing software development in Crisis," in *Procedia Computer Science*, 2019, vol. 159, pp. 1777–1786.
- [2] M. Sulayman, C. Urquhart, E. Mendes, and S. Seidel, "Software process improvement success factors for small and medium Web companies : A qualitative study," *Inf. Softw. Technol.*, vol. 54, no. 5, pp. 479–500, 2012.
- [3] G. O. Regan and I. Mackie, *Undergraduate Topics in Computer Science Concise Guide to Software Engineering Series editor*. 2017.
- [4] M. Malinova, S. Gross, and J. Mendling, "A study into the contingencies of process improvement methods," *Inf. Syst.*, vol. 104, p. 101880, 2022.
- [5] K. Petersen and C. Wohlin, "Software process improvement through the Lean Measurement (SPI-LEAM) method," *J. Syst. Softw.*, vol. 83, no. 7, pp. 1275–1287, 2010.
- [6] T. Galinac, "Empirical evaluation of selected best practices in implementation of software process improvement," *Inf. Softw. Technol.*, vol. 51, no. 9, pp. 1351–1364, 2009.
- [7] B. McFeeley, *IDEALSM : A User's Guide for Software Process Improvement. CMU/SEI-96-HB-001*, no. February. Pittsburgh: SEI, 1996.
- [8] M. B. Chrissis, M. Konrad, and S. Shrum, "CMMI: Guidelines for Process Integration and Product Improvement," 2003.
- [9] Y. T. San, *Systematic Innovation in Business & Management*. Malaysia: Firtfruit, 2019.
- [10] I. Ekmekci and E. E. Nebati, "Triz Methodology and Applications," *Procedia Comput. Sci.*, vol. 158, pp. 303–315, 2019.
- [11] K. Gadd, *TRIZ for Engineers enabling inventive problem solving*. Chichester: John Wiley & Sons, Ltd, 2011.
- [12] Y. T. San, *TRIZ Systematic Innovation in Manufacturing*. Malaysia: Firtfruit, 2018.
- [13] S. Shu Luing Nikalus, G. T. Guat, M. W. Yip, and S. C. Tai, "Systematic Innovation for Manufacturing Quality Improvement," *MATEC Web Conf.*, vol. 221, pp. 1–4, 2018.
- [14] M. Woźniak, "TRIZ Potential for IT Projects," in *Advances and Impacts of the Theory of Inventive Problem Solving*, Switzerland: Springer Nature Switzerland, 2018, pp. 169–177.
- [15] M. Belchin and P. Juberias, *Working with Functions*, no. March. 2012.