

Kajian Penerapan Aplikasi *Open Source* di Perguruan Tinggi Dengan Pendekatan *Soft System Methodology*

Studi kasus Pengembang *software* Akademik Sisfokampus

Sofian Lusa, Mario Iskandar

Laboratorium E-Government Universitas Indonesia

E-mail : sofian.lusa@ui.ac.id

1. Abstrak

Teknologi informasi diyakini sebagai salah satu domain yang strategis di dalam pembicaraan mengenai industri kreatif, karena dengan penerapan teknologi informasi yang tepat menjanjikan terciptanya nilai tambah dan keunggulan kompetitif bagi organisasi. Atas dasar ini banyak perguruan tinggi yang menggunakan *software* berbasis *closed source* ataupun *open source* sebagai bentuk kongkret pemanfaatan teknologi informasi untuk pengelolaan organisasi. Trend dalam satu dekade terakhir adalah kecenderungan institusi pendidikan untuk mencoba menggunakan *software open source* sebagai solusi ekonomis, karena adanya *constraints* dalam sumber daya dan pertimbangan kepemilikan *source code*. Penulisan artikel ini, dimotivasi masih kurangnya kajian dan studi tentang tantangan dan permasalahan penggunaan *software* berbasis *open source* di perguruan tinggi yang dikelola secara profesional. Umumnya, *software open source* yang ada dikembangkan oleh mahasiswa atau pengembang yang sifatnya sementara tanpa ada perencanaan untuk kelangsungan dalam jangka waktu panjang (*sustainability*). Fokus dari penulisan ini adalah mengidentifikasi tantangan dan permasalahan yang kongkret terjadi sebagai pembelajaran yang memberikan gambaran holistik mengenai praktek pengembangan dan penerapan *software open source*, bisnis model, dan solusi di Indonesia. Dalam kajian ini digunakan pendekatan *Soft System Methodology (SSM)* sebagai framework penyelesaian. Sedangkan obyek penelitian ditujukan kepada pengembang dari *software open source* sistem informasi akademik Sisfokampus .

Kata kunci: *open source*, kajian tantangan dan permasalahan, *Soft System Methodology (SSM)* , sisfokampus

2. Pendahuluan

Dari pengamatan penulis bahwa penerapan teknologi informasi dan komunikasi (TIK) di perguruan tinggi bukan lagi dianggap sebagai suatu pilihan tetapi sudah menjadi keharusan. Bagi perguruan tinggi yang sudah memiliki visi menjadi *cyber university*, *world class university*, atau *intelligence university* penerapan TIK merupakan hal yang fundamental untuk mencapai visi korporat. Oleh karena itu, pemanfaatan TIK secara tepat, telah menjadi alat strategis selain berfungsi untuk mempermudah atau mengatasi setiap permasalahan yang ada di dalam organisasi. Khususnya dengan tersedia perangkat lunak (*software*) akademik yang membantu proses belajar mengajar lebih fleksibel dan profesional. Persepsi bahwa pengembangan *software* membutuhkan biaya atau investasi yang besar, saat ini sudah dapat diatasi dengan adanya *software* berbasis *open source* [1]. Hadirnya *open source* ini dipicu dengan adanya kenyataan bahwa biaya atau investasi untuk penerapan *software* yang berbasis *closed source* sangat tinggi dan mendominasi perkembangan *software* dunia. Akibatnya, para pengguna (*user*) tidak banyak memiliki pilihan di dalam penerapan *software*. Fenomena gerakan *open source* dan *free software* mendapat respons yang positif dari para pengembang dan pengguna *software* di seluruh dunia. Dengan hadirnya *open source* dan *free software*, para pengguna (*user*) dapat memilih solusi alternatif dalam penerapan teknologi informasi sekaligus sebagai tandingan untuk *software closed source (proprietary)*. Di Indonesia, Gerakan *open source* di wujudkan dengan adanya gerakan IGOS (*Indonesia, Goes Open Source*) pada tahun 2004 dengan mendapat dukungan dari Menteri Riset dan Teknologi, Menteri Komunikasi dan Infomasi dan Lembaga Ilmu Pengetahuan Indonesia (LIPI). Inti dari deklarasi gerakan IGOS adalah gerakan untuk mengembangkan, mendistribusikan, dan mensosialisasikan penggunaan *software open souce* di masyarakat. Salah satu yang menjadi sasaran utama gerakan IGOS adalah institusi

pendidikan (perguruan tinggi dan sekolah). Institusi pendidikan dianggap sebagai institusi yang tepat segi pembelajaran teknologi dan dari segi kesiapan sumber daya manusia [2]. Dalam kenyataannya, untuk merealisasikan nilai tambah dalam penggunaan *software open source* faktor penentu tidak semata-mata bergantung kepada tersedianya infrastruktur teknologi informasi, dukungan kebijakan dan sumber daya, dan terpenuhinya sumber daya keuangan, namun yang perlu dicermati juga adalah faktor manajemen organisasi dan perilaku dari para pengguna. Oleh karena itu, diperlukan kajian secara holistik mengenai tantangan dan permasalahan penerapan *software open source* khususnya di perguruan tinggi sebagai pengguna. Berdasarkan fakta ini, penulis tertarik untuk melakukan kajian penerapan *software open source* di perguruan tinggi, dengan menggunakan pendekatan *Soft System Methodology* (SSM). Studi kasus yang di ambil adalah pengembang dari *software open source* sisfokampus .

3. Tujuan Dan Manfaat

Hasil dari penulisan ini berupa *model* sebagai hasil kajian yang memetakan tantangan dan permasalahan dalam penerapan *software open source* dengan menggunakan pendekatan SSM. Tujuan yang diharapkan dengan penulisan artikel ini adalah:

- a) Dapat memberikan gambaran tantangan dan permasalahan dalam penerapan *software open source*, sehingga dapat menjadi model pemikiran dan pertimbangan bagi pengguna teknologi informasi di Indonesia khususnya bagi institusi pendidikan.
- b) Sebagai salah satu pembelajaran dalam pemecahan masalah dengan pendekatan *Soft System Methodology* (SSM).
- c) Mensosialisasikan bahwa *software open source* dapat dijadikan sebagai solusi alternatif di dalam penerapan teknologi informasi.

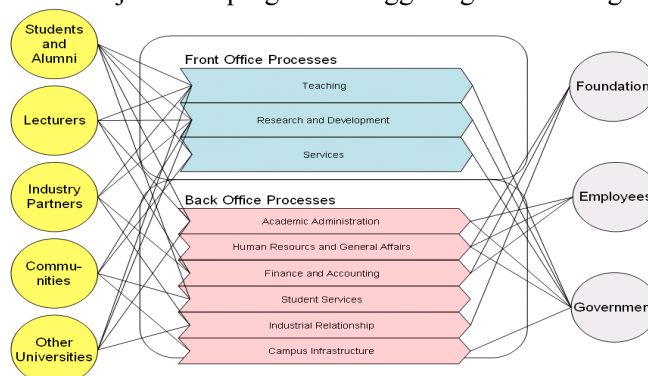
Sedangkan manfaat yang ingi diberikan sehubungan dengan penulisan ini sebagai berikut;

- a) Secara akademis, artikel ini berguna untuk menambah keragaman penerapan *software open source* dan penerapan dari model *Soft system methodology* yang membuka kesempatan untuk melanjutkan penelitian ini bagi akademisi lainnya.
- b) Bagi dunia bisnis (pengembang), mengetahui faktor-faktor kritikal tantangan dan permasalahan dalam pengembangan dan penerapan *software open source* untuk mengembangkan *software* secara efektif.
- c) Bagi pengguna (user), mengetahui faktor-faktor kritikal tantangan dan permasalahan dalam pengembangan dan penerapan *software open source* di perguruan tinggi Indonesia.

4. Pengertian Perguruan Tinggi Dan Peranan *Software*.

Pengertian perguruan tinggi adalah satuan pendidikan yang menyelenggarakan pendidikan tinggi. Perguruan tinggi menyelenggarakan pendidikan tinggi dan penelitian serta pengabdian kepada masyarakat. Penelitian merupakan kegiatan telaah taat kaidah dalam upaya untuk menemukan kebenaran dan/atau menyelesaikan masalah dalam ilmu pengetahuan, teknologi, dan/atau kesenian. Pengabdian kepada masyarakat merupakan kegiatan yang memanfaatkan ilmu pengetahuan dalam upaya memberikan sumbangan demi kemajuan masyarakat [3]. Perguruan tinggi dapat dibagi ke dalam beberapa bentuk yaitu: akademi, politeknik, sekolah tinggi, institut, dan universitas. Masing-masing dari bentuk tersebut dibedakan berdasarkan penyelenggaraan program pendidikan dan banyaknya cabang ilmu pengetahuan yang ditawarkan. Dari berbagai bentuk perguruan tinggi tersebut, jika ditinjau dari proses inti (*core process*) dari setiap perguruan tinggi adalah sama yaitu pengajaran (*teaching*), penelitian (*research*) dan pelayanan (*services*). Untuk dapat menunjang ketiga proses inti perguruan tinggi tersebut diperlukan dukungan dari sejumlah aktivitas dan proses lainnya seperti administrasi, akuntansi, keuangan, human resources, dan infrastruktur. Aktivitas dan proses pendukung dapat dikelompokkan dan dikenali dengan tujuan agar manajemen perguruan tinggi dapat fokus dalam mengalokasikan sumber daya yang tersedia secara maksimal. Untuk dapat menjalankan proses inti dan aktivitas pendukung di perguruan tinggi maka peranan *software* (piranti lunak) menjadi sangat penting dan strategis terutama dalam memenuhi dan melayani para stakeholder secara profesional. *Stakeholder* perguruan tinggi merupakan pihak yang memiliki keterkaitan erat dengan proses inti dan aktivitas penunjang dari perguruan tinggi. Setidaknya ada

delapan stakeholder perguruan tinggi yaitu mahasiswa, alumni, dosen, industri, komunitas, yayasan, karyawan, pemerintah, dan institusi pendidikan lain. Gambaran peran *software* dengan *stakeholder* dalam proses manajemen di perguruan tinggi tergambar sebagai berikut: [3]



Gambar 1: Hubungan Stakeholder dengan proses manajemen di perguruan tinggi

5. Pengertian *Software* Berbasis *Open Source* dan Sisfokampus.

Kendala yang klasik terjadi di institusi pendidikan sehubungan dengan penerapan *software* adalah terbatasnya dana mengingat investasi *software* yang berbasis *closed sources* sangat tinggi [2]. Pengertian dari *software* berbasis *closed source* adalah sistem pengembangan yang dikoordinasi oleh suatu orang/lembaga pusat, yang tidak mendistribusikan atau tidak mengizinkan sumber kode (*source code*) untuk dimanfaatkan secara utuh. Oleh karena itu *software open source* tumbuh dan diterima oleh masyarakat dunia. Latar belakang munculnya *Open source* dikarenakan para pengembang *software* (*developer*) yang berpendapat bahwa *source code* itu selayaknya dibuka terhadap publik. Agar sebuah *software* dapat bersifat *open source* ada beberapa kriteria yang harus dipenuhi. Kriteria yang terdapat dalam sebuah *software open source* di atur oleh organisasi yang bernama *Open Source Organization*. Organisasi ini yang mengeluarkan *The Open Source Definition* sebagai suatu set kondisi-kondisi yang harus dipenuhi, agar sebuah lisensi dapat disebut bersifat *open source*. Dua poin definisi yang erat kaitannya dengan bisnis adalah bahwa lisensi bersifat *open source* tidak boleh melarang pihak ketiga untuk menjual *software* berlisensi *open source* sebagai komponen dari sebuah *software* yang lebih besar, dan lisensi bersifat *open source* tidak diperbolehkan membatasi *software* lain (www.opensource.org/docs/definition.php). Dalam penulisan ini, obyek kongkret dari *software open source* yang dijadikan studi kasus adalah Sisfokampus yang telah dikembangkan sejak tahun 2003. Pemilihan Sisfokampus atas dasar alasan sebagai berikut:

- Sisfokampus terpilih untuk ditampilkan dalam Pameran Produk kreatif Indonesia (PPKI) 2009 sebagai pengakuan untuk produk kreatif dengan kategori *E-learning*.
- Sisfokampus terdaftar sebagai salah satu produk dari *Indonesia, Go Open Source* (IGOS) Bandung dan sudah diimplementasi di berbagai perguruan tinggi seperti Universitas, akademi, sekolah tinggi, politeknik, dan institut.
- Framework Sisfokampus mengacu kepada solusi *enterprise resource planning* (ERP) yang mengakomodir bisnis proses mulai dari *front office* sampai dengan *back office* serta menjamin terintegrasinya program Dikti sehingga laporan tidak dibuat secara *double entry* dan tepat waktu.
- Memiliki panduan dan manual yang lengkap sehingga memudahkan untuk mengoperasikan sesuai dengan level akses dan modul.
- Memiliki *training center* sebagai dukungan bagi perguruan tinggi yang membutuhkan pelatihan dan workshop secara intensif.

Sedangkan modul-modul yang terdapat dalam Sisfokampus, secara ringkas terdiri dari:

- Modul Administrasi Sistem, yaitu modul pengelolaan level, Modul dan Sub-modul.
- Modul Penerimaan Mahasiswa Baru (PMB), yaitu modul pengelolaan penerimaan mahasiswa baru.

- c) Modul Administrasi dan kepala Akademik, yaitu modul pengelolaan operasional akademik yang dibagi berdasarkan kepala akademik dan administrasi akademik.
- d) Modul Administrasi dan kepala keuangan, yaitu modul pengelolaan keuangan mahasiswa yang dibagi untuk kepala keuangan dan administrasi.
- e) Modul Dosen dan administrasi Dosen, yaitu modul manajemen dosen terdiri dari jadwal mengajar, perwalian, bimbingan, tugas akhir, jadwal ujian, dan cetak nilai mahasiswa .
- f) Modul Mahasiswa, yaitu modul untuk mahasiswa sehubungan dengan perkuliahan sampai proses kelulusan.
- g) Modul Alumni, yaitu modul untuk mengelola alumni.
- h) Modul Pengisian Nilai, yaitu modul untuk manajemen nilai ujian yang dapat dilakukan secara manual ataupun online.

6. Tantangan dan Permasalahan.

Kegagalan penerapan teknologi informasi telah menjadi topik bahasan dari berbagai institusi atau lembaga survei, seperti yang dilakukan oleh *Standish Group*, *IT Cortex*, *KPMG*, *Gartner*, *AMR Research*, dan *Economist intelligence*. Diantara berbagai perusahaan survei tersebut, *Standish Group* yang sering dijadikan referensi dan melakukan survei secara berkala untuk mengetahui tingkat kegagalan dari penerapan teknologi informasi (termasuk penerapan *software*). Tingkat kegagalan didefinisikan sebagai *challenged project* artinya kegagalan dalam pemenuhan waktu (*over time*), *over budget*, dan hasil yang diimplementasikan tidak sesuai dengan kebutuhan dan *impaired project*, artinya proyek tersebut dibatalkan. Gambaran ringkas dan penyebab kegagalan dari penerapan teknologi informasi adalah: [4]

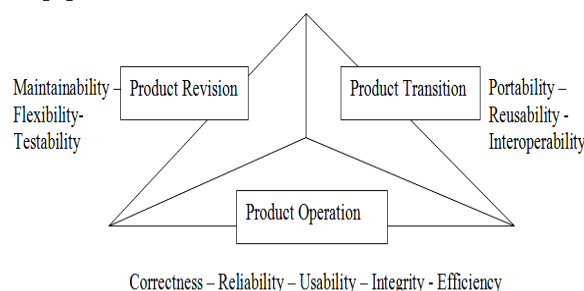
Factor	Challenged Projects	Impaired Projects
Incomplete Requirements & Specifications	12.3%	13.1%
Lack of User Input	12.8%	12.4%
Changing Requirements & Specifications	11.8%	8.7%
Lack of Resources	6.4%	10.6%
Lack of Executive Support	7.5%	9.3%
Unrealistic Expectations	5.9%	9.9%
Technology Incompetence / Illiteracy	7.0%	4.3%
Lack of Planning	0.0%	8.1%
Didn't Need It Any Longer	0.0%	7.5%
Lack of IT Management	0.0%	6.2%
Unclear Objectives	5.3%	0.0%
Unrealistic Time Frames	4.3%	0.0%
New Technology	3.7%	0.0%
Other	23.0%	9.9%

Gambar 2: *Standish Group Report*

Berdasarkan pengalaman dan pengamatan dari *developer software Sisfokampus* dapat ditentukan bahwa tantangan dan permasalahan sehubungan penerapan *software open source* sebagai berikut:

- a) Tidak lengkapnya kebutuhan dan spesifikasi (*incomplete requirement and spesification*)
- b) Perubahan kebutuhan dan spesifikasi (*changing requirement and spesification*)
- c) Terbatasnya sumber daya (*Lack of resources*)
- d) Ekspektasi yang tidak realistis (*unrealistic expectations*)
- e) Kurang komitmen dalam manajemen (*Lack of IT Management*)

Dari kelima komponen diatas, disimpulkan bahwa tantangan dan permasalahan yang dihadapi oleh *developer* berhubungan erat dengan manajemen, organisasi, dan manusia. Dari sisi lain, jika dikaitkan dengan kualitas *software*, maka terdapat tiga komponen penting yang harus dimiliki oleh *software* berkualitas yaitu [5]:

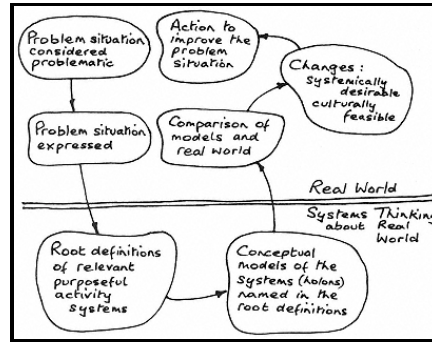


Gambar 2: *Quality of Software*

1. *Product Operations*. Faktor – faktor operasional dari *software* yang dimaksud sebagai berikut:
 - a) *Correctness* - sejauh mana suatu *software* memenuhi spesifikasi dan *mission objective* dari users;
 - b) *Reliability*- sejauh mana suatu *software* dapat digunakan dalam melaksanakan fungsinya sesuai dengan requirement yang dibutuhkan.
 - c) *Efficiency*- seberapa banyaknya sumber daya programmer dan coding program yang dibutuhkan suatu *software* untuk melakukan fungsinya;
 - d) *Integrity* – sejauh mana kontrol keamanan *software* dan data terhadap pihak yang tidak berhak menggunakannya.
 - e) *Usability*- usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari *software*.
2. *Product Revision*. Faktor-faktor yang berkaitan dengan kemampuan *software* untuk menjalani perubahan adalah sebagai berikut:
 - a) *Maintainability*- usaha yang diperlukan untuk menemukan dan memperbaiki kesalahan terhadap *software*.
 - b) *Flexibility*- usaha yang diperlukan untuk melakukan modifikasi terhadap operasional *software*.
 - c) *Testability*- usaha yang diperlukan untuk menguji suatu *software* untuk memastikan apakah melakukan fungsi yang dikehendaki atau tidak.
3. *Product Transition*. Faktor-faktor yang berkaitan dengan tingkat kemampuan beradaptasi *software* terhadap lingkungan baru, yaitu :
 - a) *Portability*- usaha yang diperlukan untuk mentransfer *software* dari suatu hardware dan sistem tertentu agar dapat berfungsi pada hardware dan sistem lainnya.
 - b) *Reusability*- sejauh mana suatu *software* (atau bagian *software* dapat dipergunakan kembali pada pembuatan *software* lainnya.
 - c) *Interoperability*- usaha yang diperlukan untuk menghubungkan suatu *software* dengan *software* lainnya.

7. Metodologi

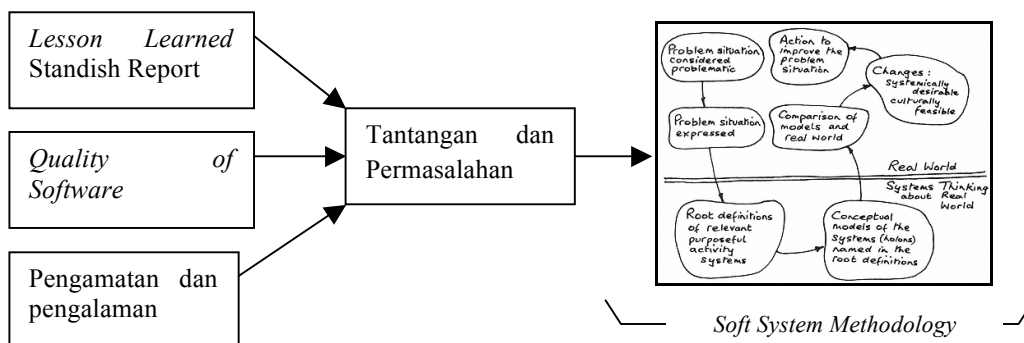
SSM merupakan metodologi sebagai pembaharuan dari *Hard System methodology (HSM)* yang pola pikirnya adalah membatasi jumlah variabel seminimum mungkin sehingga dapat menyederhanakan masalah dan memudahkan perumusan formulasi solusi. Kelemahan dari HSM adalah tidak cocok digunakan untuk permasalahan organisasional oleh karena ketika formulasi solusi telah berhasil dirumuskan, formula tersebut hanya mampu menjawab permasalahan sebatas pada saat itu saja kemudian tidak akan relevan lagi ketika waktu berjalan dan salah satu unit variabel dalam domain masalah (organisasi) mengalami perubahan. Untuk dapat membuat kajian penerapan *software open source* ini, digunakan *Soft system methodology (SSM)*. Pertimbangannya persepektif manajerial dan organisasi bersifat kompleks (*complicated*) dan dinamis cenderung berubah-ubah mengikuti perkembangan teknologi. Metode SSM tidak membatasi permasalahan pada variabel tertentu saja namun mencoba mengidentifikasi sebanyak mungkin aspek (variabel) yang berinteraksi di dalam sistem. Dengan demikian pendefinisian permasalahan akan lebih lengkap karena mempertimbangkan banyak aspek dan mampu mengantisipasi kemungkinan perubahan (dinamika) yang akan terjadi. Pada akhirnya solusi yang nantinya akan dirumuskan dapat lebih efektif dan relevan dengan kondisi real organisasi (internal/eksternal). Ada tujuh langkah dalam menggunakan kerangka SSM yaitu: [6]



Gambar 3. Kerangka Kerja SSM

1. Situasi permasalahan (*problem situation*), yaitu mulai mengenali situasi dan permasalahan yang sedang terjadi pada domain yang sedang diobservasi.
2. Penggambaran situasi permasalahan kedalam diagram *rich picture* (*problem situation expressed*), yaitu menggambar sketsa situasi real permasalahan kedalam sebuah diagram *rich picture* yang besar (*helicopter view*).
3. Pendefinisian kata-kata kunci (*root definitions*), yaitu mulai mengumpulkan kata-kata kunci yang harus didefinisikan masing-masing ke dalam bentuk jalan cerita proses bisnis secara tektual dan ringkas. Dari *Root Definition* ini dipetakand ke dalam elemen **CATWOE**- (*Client, Actor, Transformation, World view, owner, environment*).
4. Pembuatan model sistem berdasarkan *root definitions* (*conceptual modeling*), untuk setiap definisi dibuatkan sebuah diagram model dalam bentuk diagram *rich picture*.
5. Membandingkan model dengan situasi sesungguhnya (*comparison of models and real world*), yaitu melakukan perbandingan antara sketsa situasi riil dengan model yang dibuat.
6. Melakukan perubahan/penyesuaian (*changes*), jika ada perbedaan maka dilakukan penyesuaian-penyesuaian hingga model konseptual sudah sesuai dengan situasi riil.
7. Melakukan perbaikan/solusi untuk sistem yang direkomendasikan (*action to improve the problem situation*), fase akhir adalah melakukan rekomendasi-rekomendasi perbaikan terhadap sistem yang lama.

Metodologi yang digunakan dalam melakukan kajian ini, terlihat seperti dibawah ini:



Gambar 4 : Metodologi Penelitian

8. Perancangan Kajian dengan SSM

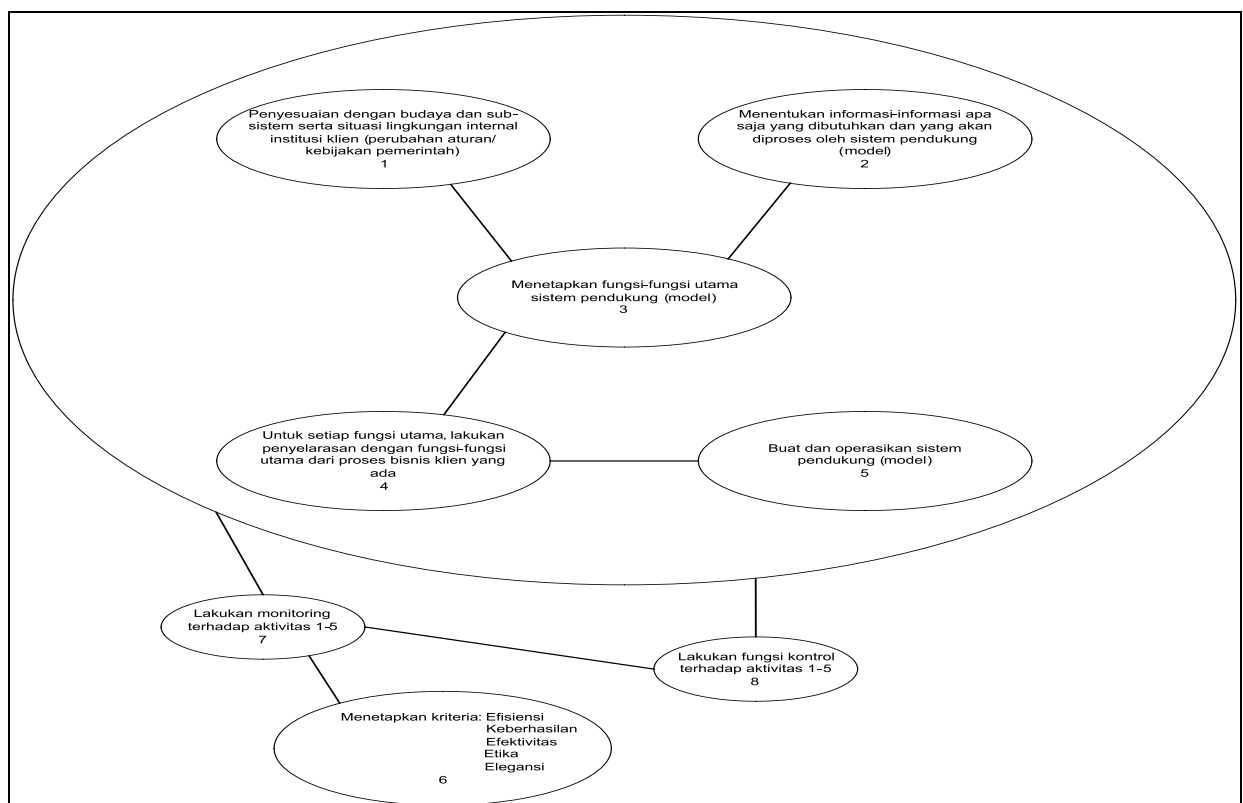
Diakui bahwa secara umum penerapan *software open source* belum cukup serius dilakukan khususnya pada institusi pendidikan swasta maupun negeri. Hal ini cukup wajar mengingat belum begitu banyaknya pilihan *software open source* yang berkualitas sekarang ini. *Software Sisfokampus* merupakan sistem dengan platform yang terbuka sehingga bebas dikembangkan sendiri (*in-house development*) dan tidak dibebankan biaya untuk kode sumbernya. Tantangan dan permasalahan dalam penerapan *software Sisfokampus* yang *open source* selama ini mendorong inisiatif untuk membuat suatu kajian dengan tujuan membuat sebuah model yang mampu mendukung proses penerapan yang lebih baik khususnya bagi *software opensource* lainnya.

Sedangkan analisa dalam elemen **CATWOE-** (*Client, Actor, Transformation, World view, owner, environment*) adalah sebagai berikut:

- C : Institusi perguruan tinggi (*Problem Domain*)
- A : Semua level staf (*top-mid-operational*)
- T : Kebutuhan akan *model* yang mampu menerima, menyimpan, menganalisa dan membangun suatu sub-sistem baru (*customation*) yang sesuai kebutuhan perguruan tinggi (klien).
- W : Model ini diklaim mampu mendukung penerapan Sistem Sisfokampus yang lebih baik
- O : Vendor SisfoKampus, Institusi Klien (*Problem Domain*)
- E : Kebijakan dan Aturan Pemerintah tentang Pendidikan serta Perubahan-perubahannya.

8.4 Conceptual Model

Gambar model konseptual awal yang dihasilkan dari *root definition* adalah “**sebuah sistem pendukung (model) penerapan aplikasi opensource Sisfokampus**” dengan memperhatikan faktor efisien, keberhasilan, efektivitas, etika dan elegansi adalah sebagai berikut:



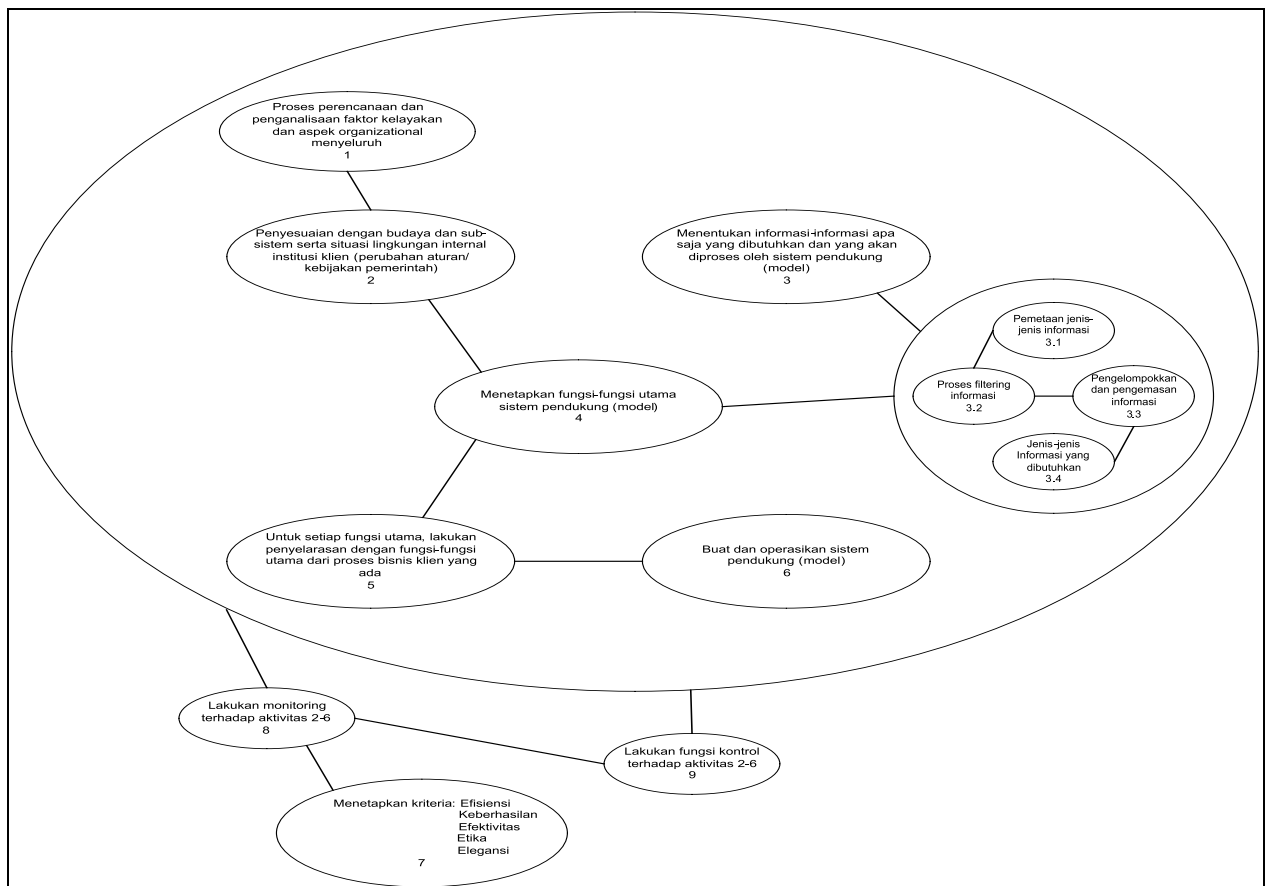
Gambar 6: *Conceptual Model* yang Diturunkan dari *Root Definition* Sistem

- Efisiensi: *Outcome/output* dibagi dengan sumber daya yang digunakan
- Keberhasilan: Sistem berjalan dengan sebagaimana mestinya.
- Efektivitas: Mampu memberikan kontribusi yang langsung dan signifikan kepada peningkatan kinerja bisnis secara jangka panjang
- Etika: Manajemen dan fungsi kontrol terhadap jenis-jenis data, pembagian tata-cara perlakuan berdasarkan sifat atau jenis data (seperti fungsi redaksional dan moderatori)
- Elegansi: Tata-cara pengelolaan operasional fungsi-fungsi utama sistem yang dapat berjalan selaras/harmonis terhadap fungsi-fungsi utama dari proses bisnis

8.5 Comparison Conceptual Model and Real World.

Selanjutnya untuk dapat merealisasikan *conceptual model* agar lebih holistik dan sesuai dengan situasi permasalahan saat ini (*real world* dari *rich picture*). Ada 3 kelompok besar aktivitas utama yang harus

ada didalam suatu sistem yaitu: perencanaan, eksekusi dan pengecekan. Model konseptual awal belum memiliki aktivitas perencanaan sehingga kemudian ditambahkan satu aktivitas lagi (no. 1, perencanaan dan penganalisaan) sebelum aktivitas penyesuaian dengan kondisi perusahaan. Aktivitas perencanaan dan penganalisaan ini meliputi kegiatan studi kelayakan, analisa sumber daya yang sudah dimiliki serta analisa menyeluruh untuk aspek organisasional. Disamping itu, pada aktivitas penentuan jenis-jenis informasi untuk diolah dan digambarkan sub-sub aktivitas inti agar lebih jelas informasi yang akan diolah oleh sistem itu sudah benar-benar valid (berdasarkan prinsip GIGO-*garbage in garbage out*). Sub-sub aktivitas inti ini adalah:



Gambar 7: Model Konseptual Perbaikan

- Pemetaan jenis-jenis informasi, dipetakan antara seluruh informasi saat ini yang digunakan/dimiliki dengan informasi yang saat ini belum dapat dimiliki (harapan) sehingga diperoleh *gap*/kesenjangan informasi.
- Penyaringan informasi, berdasarkan hasil pemetaan kemudian akan disaring (*filtering*) informasi mana saja yang benar-benar dibutuhkan untuk membantu pemecahan masalah saat ini.
- Pengelompokan dan pengemasan jenis-jenis informasi, informasi-informasi yang sudah berhasil disaring kemudian akan dikategorisasi sedemikian rupa sehingga akan mempermudah dalam proses pencarian dan penelusuran (*inquiry process*).
- Output* yang berupa kelompok-kelompok informasi yang dibutuhkan, hasilnya (*output*) inilah yang akan menjadi input ke sistem untuk disimpan, diolah dan disebar (fungsi utama sistem) kesemua *stakeholders* yang membutuhkan.

8.6. Kebutuhan Perubahan Dan Strategi Pendukung (Model)

Setelah dilihat dari analisa dan hasil analisa: *rich picture*, model konseptual awal dan model konseptual terhadap sistem pendukung (model), maka ada beberapa kebutuhan dan perubahan kritical yang harus diantisipasi oleh model tersebut. Pada tahap meminta masukan dari internal *stakeholder* utama sudah terlihat dan tergambar (pada model konseptual hasil perbaikan) ada penambahan 1 aktivitas perencanaan dan 4 sub-aktivitas dari aktivitas no. 3 yaitu penentuan jenis-jenis informasi.

8.7. Rencana Perbaikan atau Solusi Atas Sistem Yang Lama

Berdasarkan hasil temuan tersebut maka di rancang *action plan* sebagai solusi atas perubahan yang dilakukan dalam rangka memperbaiki sistem pendukung (model) dalam penerapan Sisfokampus, yaitu:

- a) Harus ada proses perencanaan dan penganalisaan yang baik atas faktor-faktor kelayakan dan aspek organizational secara menyeluruh (*thorough*).
- b) Harus ada proses yang menentukan informasi-informasi apa saja yang dibutuhkan dan relevan yang nantinya akan diproses oleh sistem pendukung (model) dalam penerapan sistem Sisfokampus . Proses ini terdiri dari beberapa langkah ,yaitu:
 - a. Pemetaan jenis-jenis informasi.
 - b. Proses penyaringan (*filtering*) informasi.
 - c. Pengelompokkan dan pengemasan informasi.
 - d. Jenis-jenis informasi yang dibutuhkan dan paling relevan.
- c) Dan, harus ada proses manajemen proyek yang baik (kontrol dan pengawasan) selama proses-proses inti dilakukan agar sesuai analisa, rencana dan target yang telah dibuat sebelumnya (langkah ke-1).

9. Kesimpulan

Kajian model dengan menggunakan SSM menjadi salah satu alternatif untuk memecahkan tantangan dan permasalahan yang dihadapi oleh *developer software open source* dalam mengembangkan produknya. Untuk dapat secara efektif menggunakan SSM, terlebih dahulu harus ditetapkan permasalahan yang terjadi dan setelah itu dipetakan ke dalam langkah-langkah di dalam SSM. Dengan menggunakan SSM, diharapkan permasalahan dan solusi yang dihasilkan akan lebih holistik sehingga dapat diakomodir berbagai perspektif yang ada dan terjadi di dalam lingkungan yang sedang dihadapi.

DAFTAR PUSTAKA:

- [1] Nugroho, Lukito E (2005).*Model Open source dan pembelajaran TI di sekolah di Indonesia*, Diakses tanggal 28 Februari 2010 dari www.mti.ugm.ac.id/~lukito/.../OS%20di%20Sekolah%20Menengah.ppt
- [2] Wahid, Fathul (2004). *Peluang dan tantangan pemanfaatan teknologi informasi di perguruan tinggi*. Diakses tanggal 28 Febrauri 2010 dari <http://journal.uui.ac.id/index.php/media-informatika/article/view/2/2>
- [3] Indrajit, Eko (2004), *Manajemen Perguruan Tinggi Modern*, e-book
- [4] Ray, Samir & Patel, Dipesh (2008), *Managing Chaos in an Agile world*, PM World Today, November 2008 (Vol. X, Issue XI)
- [5] Roger S,Pressman (2005). *Software Engineering*, Mc Graw-Hill, International Edition.
- [6] Checkland, Peter, Scholes, Jim. (1999) “*Soft Systems Methodology in Action: A 30 Years Retrospective*”, Inggris: John Wiley.