LAPORAN PENELITIAN HIBAH MONODISIPLIN

PENGEMBANGAN MODEL DAN *TOOLS*

UNTUK MENGAUDIT APLIKASI WEB SEKALA BESAR

(*WEB ENTERPRISE AUDIT*)

Disusun Oleh:

Dr. Veronica S. Moertini, Ir., M.T. (Ketua)
Gede Karya, S.T., M.T. (Anggota)

Lembaga Penelitian dan Pengabdian kepada Masyarakat
Universitas Katolik Parahyangan
2013

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan bimbingan-Nya penelitian ini dapat diselesaikan dengan baik. Penelitian ini merupakan rangkain dari penelitian dalam rangka mengembangkan metode dan *tools* untuk melakuan audit terhadap aplikasi web sekala besar (*web enterprise*).

Pada kesempatan ini, kami menyampaikan terima kasih yang sebesar-besarnya, kepada Ketua Jurusan Teknik Informatika dan Dekan Fakultas Teknologi Informasi dan Sains (FTIS) atas ijin dan dukungannya sehingga penelitian ini dapat terselenggara. Terima kasih juga kami sampaikan kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Unpar, atas fasilitasi dan dukungan dana untuk menyelesaikan dan diseminasi penelitian ini pada *Information System International Conference* (ISICO) yang ke-2 di Denpasar, Bali pada awal Desember 2013 yang lalu.

Penelitian ini juga dapat diselesaikan atas kerjasama yang baik antara tim peneliti, dan dengan mahasiswa yang membantu proses pengembangan dan pengujian *tools* IWTE, serta staf penunjang FTIS yang mendukung administrasi dan fasilitasi.

Kami berharap hasil penelitian ini bermanfaat, berkontribusi pada aspek ilmu audit sistem informasi, maupun pengembangan *tools* yang digunakan dalam audit aplikasi web sekala besar yang saat ini marak digunakan dalam bisnis maupun komunitas. Dengan audit yang memadai, kami berharap kualitas layanan dan operasi aplikasi web sekala besar menjadi lebih baik.

Demikian kami sampaikan, mohon maaf jika dalam pelaksanaan penelitian maupun dalam penyajian laporan ini ada hal-hal yang kurang berkenan.

Terima kasih.

Tim Peneliti

DAFTAR ISI

ABSTRAK

Pemanfaatan aplikasi web sekala besar (*web enterprise*) untuk mengimplementasikan sistem informasi pada organisasi memberikan manfaat (benefit) dan potensi resiko. Sebagai proses pengendalian resiko, perlu dilakukan audit secara berkala. Agar audit berjalan efektif, diperlukan metodologi, model audit dan kakas (*tools*).

Pada penelitian ini diusulkan model audit aplikasi web sekala besar (*web enterprise audit*) dengan mengkombinasikan framework ISACA dan ISO/IEC 25010. Framework ISACA lebih berfokus pada aspek fungsional dan tata kelola, sedangkan ISO/IEC 25010 berfokus pada aspek non fungsional, khususnya performansi dan reliabilitas yang mencerminkan karakteristik aplikasi web sekala besar. Model yang diusulkan diharapkan menyediakan jaminan kualitas yang lebih komprehensip.

Untuk mengimplementasikan model usulan, kakas uji *Integrated Web Test Environment* (IWTE) yang telah dihasilkan pada penelitian sebelumnya diadopsi dan dikembangkan lebih lanjut. IWTE telah menyediakan modul uji fungsional dan uji beban. Agar memenuhi kebutuhan model usulan, pada penelitian ini, IWTE dikembangkan dengan menambah modul uji pengendalian yang diturunkan dari *application control (AC) assesment* dari COBIT 4.1 yang digunakan dalam framework ISACA.

Hasil pengujian pada kasus Student Portal Unpar menunjukkan bahwa model dan kakas yang diusulkan dapat memberikan hasil yang memadai. Namun, pada proses pengujian ditemukan kesulitan dalam *deployment* aplikasi agen uji IWTE khususnya untuk uji beban. Untuk penelitian lanjutan, diberikan saran untuk mengembangkan aplikasi IWTE dengan kelengkapan fungsi analisis statistik dan visualisasi hasil pengujian.

Kata kunci: audit aplikasi web sekala besar, web enterprise audit, IWTE, framework ISACA, ISO/IEC 25010

BAB I. PENDAHULUAN

Pemanfaatan web enterprise dalam mengimplementasikan sistem informasi memberikan manfaat (benefit) dan potensi resiko. Sebagai proses pengendalian resiko, perlu dilakukan audit secara berkala. Agar audit berjalan efektif, diperlukan metodologi dan *tools*. Dalam audit diperlukan uji kepatuhan (*compliance*) dan uji substantif (*substantive*). Pada uji substantif perlu dilakukan pengujian sistem secara langsung, untuk membuktikan pemenuhan pada aspek fungsional maupun aspek non fungsional, seperti: performansi dan kapasitas.

Penelitian ini merupakan bagian dari rangkaian penelitian dalam mengembangkan perangkat lunak bantu untuk mengaudit sistem informasi berbasis web sekala besar (*web enterprise*). Pada penelitian 1-3 telah dikembangkan perangkat lunak untuk melakukan otomasi uji fungsional [8] dan uji non fungsional berbasis web [7]. Pada penelitian ke-4 semua perangkat lunak tersebut telah diintegrasikan ke dalam satu situs *Integrated Web Test Environment* (IWTE) yang dapat diakses di http://webtest.unpar.ac.id. Penelitian ini akan difokuskan pada mengadopsi dan mengadaptasi standar audit sistem informasi dari ISACA/ COBIT untuk mengaudit sistem informasi berbasis web sekala besar, dan menggunakan IWTE sebagai kakas uji substantif. Hasil yang diharapkan berupa model dan perangkat lunak *tools* yang berisi ceklist dan kakas uji substantif, serta contoh penerapannya pada sistem informasi berbasis web.

Rumusan masalah dalam penelitian ini diformulasikan dalam bentuk 2 pertanyaan berikut:

1. *Bagaimana melakukan audit aplikasi web sekala besar?*
    Dalam hal ini perlu dicari metodologi yang tepat sehingga dapat dipertanggungjawabkan, serta mendefinisikan kakas uji yang diperlukan. Metodologi diadopsi dan diadaptasi dari standar audit sistem informasi yang berlaku umum. Hasil adopsi berupa model audit aplikasi web sekala besar.

2. *Bagaimana mengadopsi dan mengadaptasi IWTE sebagai tools audit aplikasi web sekala besar?*
    Hal ini perlu agar proses pengujian, khususnya substantif dapat dilaksanakan dengan meyakinkan. Juga mendukung aspek sekalabilitas dari kakas uji untuk menghasilkan trafik data yang diperlukan. Adaptasi diperlukan untuk mendukung aspek-aspek yang tidak dicakup oleh IWTE, khususnya ceklist sebagai kertas kerja dalam audit.

Dengan demikian ruang lingkup penelitian mencakup:

1. Mengembangkan model audit *web enterprise* dengan mengadopsi dan mengadaptasi standar dari ISACA/ COBIT.
2. Membuat panduan dan kakas audit menggunakan IWTE.
3. Mengujicobakan penggunaan panduan dan kakas pada kasus tertentu.

Hasil yang diharapkan dari penelitian ini adalah:
1. Dokumentasi model dan *tools* untuk mengaudit aplikasi web sekala besar.
2. Hasil penerapan pada studi kasus

BAB II. TINJAUAN PUSTAKA

Pada bagian ini dijelaskan tentang tinjauan pustaka, yang mencakup: audit sistem informasi dari sumber ISACA sebagai dasar bagi audit aplikasi web sekala besar, kemudian dilanjutkan dengan penelitian sejenis, dan pembahasan khusus tentang IWTE.

2.1. Audit Sistem Informasi dan Aplikasi

Pada bagian ini dibahas tentang audit sistem informasi dari ISACA. *Information System Audit and Control Association* (ISACA), merupakan asosiasi profesi audit dan keamanan sistem informasi internasional yang berpusat di Amerika, dan memiliki *chapter* hampir di setiap negara, termasuk Indonesia. ISACA telah me-*release* standar dan panduan untuk melakukan audit teknologi informasi (*IT Audit)* [1]. Di dalam standar tersebut dibahas tentang *Code of Profesionals Ethics*, dan *IT Audit and Assurance Standars and Guidelines*. Standar tersebutlah saat ini dijadikan sebagai acuan dalam melakukan audit terhadap sistem-sistem berbasis teknologi informasi.

Selain standar, juga disediakan bunga rampai yang berisi kodifikasi pengetahuan yang diadopsi dari berbagai *best practice* dan standar teknologi informasi, keamanan dan pengendalian. Bunga rampai ini dikodifikasi dan distrukturkan menjadi sebuah kerangka kerja (*framework*) *Control Objectives for Information and related Technology* (COBIT®) [2]. Untuk setiap panduan audit di ISACA selalu dikaitkan dengan *knowledge* yang ada di dalam COBIT. Saat ini *release* terakhir dari COBIT adalah versi 5.0 pada bulan Mei 2012 namun kondisinya belum lengkap. *Release* lengkap yang terakhir adalah versi 4.1 pada tahun 2007. Oleh karena itu, versi yang banyak digunakan saat ini adalah 4.1 [2].

Dalam melakukan audit, ada 2 kriteria yang dinilai, yaitu aspek kepatuhan (*compliance*) dan aspek substantif. Aspek kepatuhan dikaitkan dengan pemenuhan akan peraturan (legal), kebijakan, standar dan *best practice* yang ada. Sedangkan aspek substantif terkait dengan aspek efektifitas penerapan (aplikasi) pengendalian pada setiap elemen sebagai turunan dari proses bisnis yang diotomasi oleh teknologi informasi. Berbagai aspek dalam pemenuhan kepatuhan teknologi informasi telah dibahas [3].

Untuk menilai tingkat pengendalian digunakan konsep *maturity* (kematangan) yang diadopsi dari model kematangan pada perangkat lunak. ISACA juga menyediakan panduan dan alat untuk melakukan assesment terhadap kematangan pengendalian ini [4][5]. Untuk melakukan pengujian substantif terhadap program aplikasi, telah diterbitkan panduan umum audit untuk aplikasi [6] dan diuraikan lebih lanjut oleh Tommie (2012) pada artikel [7] dan [8].

2.2. Tinjauan Penelitian Sejenis

Penerapan teknologi informasi pada sektor bisnis, khususnya aplikasi web menimbulkan adanya perubahan paradigma, teknik dan *tools* dalam melakukan audit. Ada beberapa publikasi yang telah ditinjau oleh peneliti dalam hal ini dalam rentang periode 2001 – 2012.

Zabihollah Rezaee, dkk (2001) [9] telah meneliti bahwa pada saat teknologi *e-commerce* dan internet diterapkan, maka telah terjadi perubahan praktik bisnis, proses penyimpanan dan pemrosesan transaksi bisnis (*e-business*). Untuk itu, auditor perlu melakukan audit terus menerus, untuk menjamin pengendalian internal. Abu Musa (2004) [10] menyatakan bahwa audit *e-business* merupakan tantangan bagi eksternal auditor. Kemudian Junaid M. Shaikh (2005) [11] juga menyatakan bahwa pada saat bisnis suatu organisasi bertumpu pada teknologi informasi, maka auditor perlu menerapkan kerangka kerja EA (*electronic auditing*) yang bersesuaian dengan teknologi yang digunakan. Penerapan EA bagian dari *computer-assisted auditing techniquies* (CAATs).

Penerapan *e-business* menggunakan aplikasi web, telah mendorong para ahli auditor untuk mengantisipasi resiko dan pengendaliannya sehingga menjamin keamanan dan efektifitas bisnis. Susan [12], telah mengidentifikasi ada 10 resiko dari aplikasi web, diantaranya: (1) *otentification*, (2) *session security and session IDs*, (3) *SQL injection*, (4) *Buffer overflows*, (5) *Cross-site scripting (XSS)*, (6) *error handling*, (7) *remote web site administration*, (8) *denial of service*, (9) *storage*, (10) *web application testing*. Untuk setiap resiko juga telah diidentifikasi *best practice* untuk mengendalikannya. Khusus untuk (10) disarankan menggunakan lingkungan uji terintegrasi untuk secara reguler melakukan pengujian, dan jika ada perubahan yang dilakukan.

Pada aspek keamanan aplikasi web, Popa (2009) [13] juga telah meneliti bagaimana mendeteksi celah keamanan pada aplikasi web. Hal ini penting, karena penerapan aplikasi web yang kompleks meningkatkan resiko keamanan suatu bisnis. Untuk menangani keamanan dari sisi tata kelola, Colin Wathson (2009) [14] telah mengusulkan bagaimana menerapkan *COBIT Security Baseline* pada aplikasi bisnis berbasis web. Pada penelitian ini disimpulkan bahwa COBIT dapat memandu auditor untuk membuat prioritas pengendalian sehingga menurunkan resiko keamanan aplikasi web. Hal ini sebagai bagian dari tata kelola aplikasi web pada sektor bisnis.

Dari sisi *performance* dan *load*, Minesce (2002) [15] telah mengusulkan model uji beban web site menggunakan konsep emulasi kondisi peak season dengan mempertimbangkan prilaku user. Model yang diusulkan oleh Minesce banyak diadopsi oleh kalangan industri untuk membuat produk propietary-nya sendiri. Seng (2009) [16] menyatakan bahwa model benchmark di dunia industri yang digunakan untuk mengukur kinerja (*performance*) aplikasi seperti TREC, TPC, SPEC, SAP, Oracle, Microsoft, IBM, Wisconsin, AS^{3}AP, OO1, OO7, XOO7 terlalu spesifik untuk sistem tertentu saja (propietary), sehingga tidak mudah diterapkan pada aplikasi berbasis web lainnya. Untuk itu, dikembangkan model "*generic construct based workload*" yang diturunkan dari spesifikasi kebutuhan, kemudian ditranslasi menjadi 3 model, yaitu: *page model, query model dan control model*.

Dalam melakukan pengujian aplikasi web, Sampath (2012) [17] meneliti bagaimana meningkatkan efektifitas pengujian aplikasi web yang menggunakan *user-session-based*. Pada penelitian tersebut disimpulkan bahwa pengurutan dapat mengurangi test suite sehingga meningkatkan efektifitas pengujian aplikasi web yang memiliki karakteristik tersebut.

Dalam melakukan audit aplikasi, Tommie (2012) [7] telah mendefinisikan kerangka kerja untuk mengaudit aplikasi. Dalam audit ini digunakan framework berbasis proses yang terdiri atas beberapa langkah, antara lain: (1) *Plan the audit*, (2) *Determine audit objectives*, (3) *Map systems and data flows*, (4) *Identify key controls*, (5) *Understand application's functionality*, (6) *Perform applicable*

tests, (7) *Avoid/consider complications*, (8) *Include financial assertions*, (9) *Consider beneficial tools*, (10) *Complete the report*. Pada [8], Tommie menyimpulkan bahwa keberhasilan dari audit aplikasi tegantung pada reliabilitas pendekatan, dan alat yang digunakan, khususnya untuk proses (3) dan (6) khususnya CAATS.

Dalam rangka mengembangkan model dan *tools* untuk melakukan otomasi pengujian dalam sekala besar, Gede Karya (2011-2012) [18][19] telah mengembangkan alat uji performansi dan fungsional dengan menerapkan konsep agen cerdas terdistribusi pada lingkungan jaringan komputer tersebar. Lebih jauh diuraikan pada bagian 2.3.

2.3. Integrated Web Test Environment (IWTE)

Dalam rangka mengembangkan perangkat lunak uji terintegrasi untuk aplikasi web sekala besar (*web enterprise*), telah dilaksanakan serangkaian penelitian dengan dana dari LPPM-Unpar. Roadmap penelitian ini dapat dilihat pada Gambar 2.1.
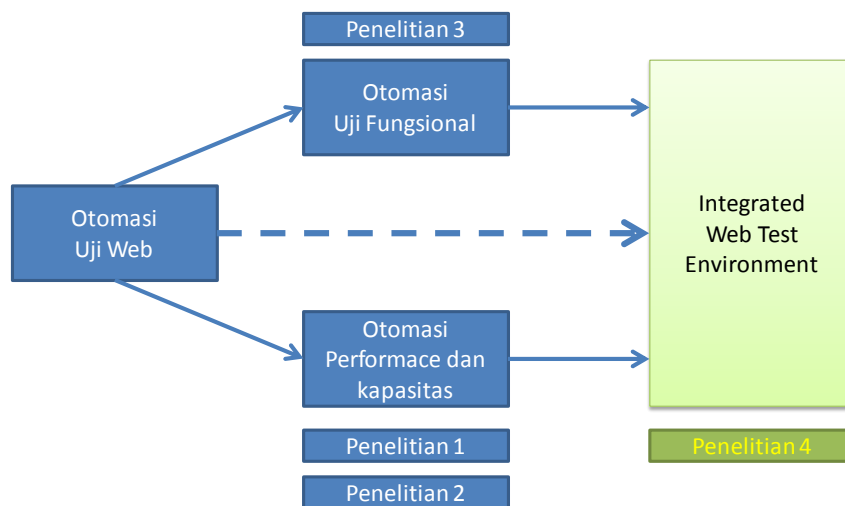
Gambar 2.1. *Roadmap* Penelitian IWTE

Rangkaian penelitian pada Gambar 2.1 didasari pada prinsip, bahwa dalam rangka menjamin kualitas aplikasi web, diperlukan adanya pengujian. Pengujian mencakup uji fungsional, dan uji non fungsional [20]. Pengujian fungsional fokus pada aspek otomasi proses bisnis (fungsi). Sedangkan pengujian non fungsional, ditekankan pada aspek kapasitas/ *throughput* dan *performance*. Karena konteks aplikasi web enterprise, maka diasumsikan bahwa aplikasi yang diuji memiliki fungsi yang banyak dengan kompleksitas yang tinggi, sehingga jika dilakukan secara manual akan memakan sumberdaya yang besar. Oleh karena itu, dikembangkan perangkat lunak untuk mengotomasi proses uji kapasitas dan uji fungsional tersebut. Untuk mengemulasi agen uji yang banyak, maka digunakan prinsip aplikasi telematika, dimana dikembangkan agen-agen uji virtual yang saling bekerjasama dalam melakukan pengujian dalam sekala besar [21].

Pada Gambar 2.1, penelitian dimulai dengan mengembangkan perangkat lunak untuk otomasi uji performansi. Pada penelitian ini digunakan metode multi agent terdistribusi [22] untuk mengemulasi agen uji dalam jumlah banyak pada lingkungan jaringan komputer. Model logika dari penelitian ini dapat dilihat pada Gambar 2.2.
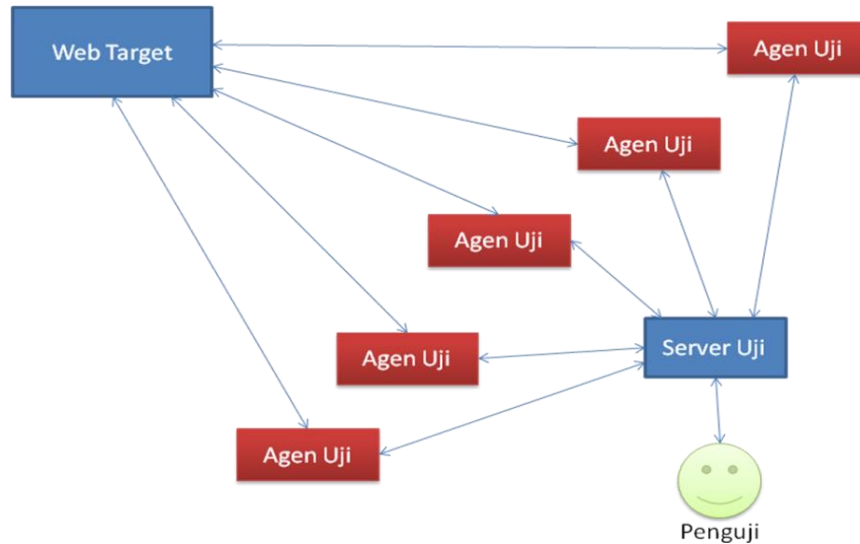
Gambar 2.2. Model Otomasi Multi Agen Terdistribusi

Pada Gambar 2.2 dapat dilihat bahwa penguji memberikan instruksi melalui Server Uji, kemudian Agen Uji menjalankan instruksi tersebut untuk melakukan uji kapasistas dan performansi ke web target. Model ini dapat mensimulasikankan kondisi peak season yang menghasilkan trafik dalam jumlah besar untuk uji kapasitas web target (beban). Model tersebut diimplementasikan dengan konsep multi threading [23] dengan bahasa Java pada lingkungan J2SE [24] seperti pada Gambar 2.3.
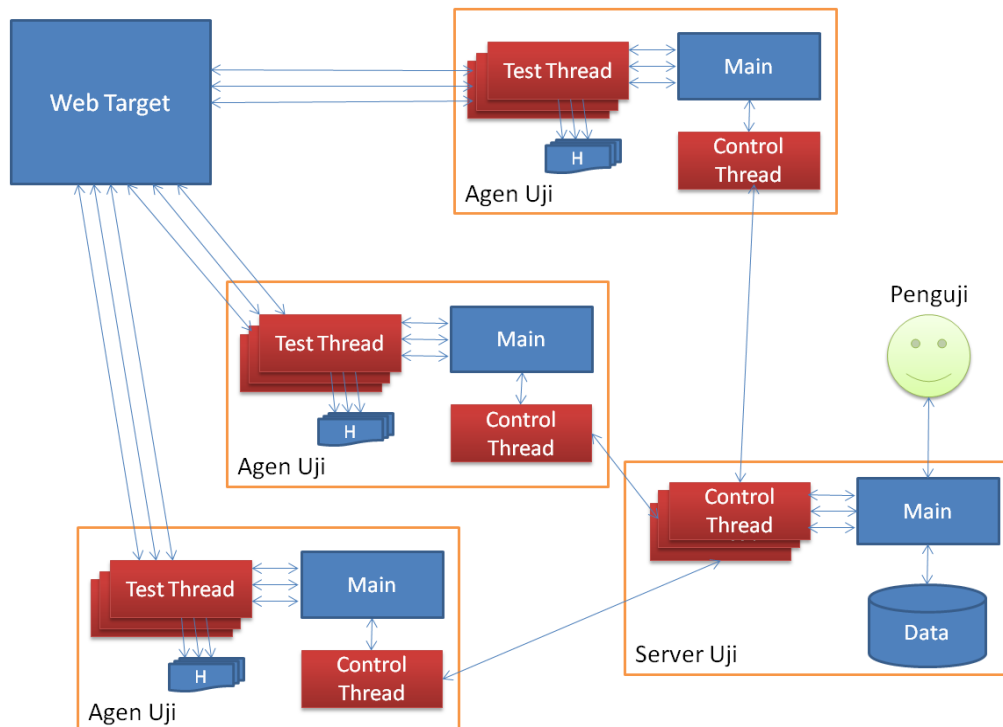
Gambar 2.3. Implementasi Model Otomasi Multi Agen dengan Multithreading

Dengan implementasi pada Gambar 2.3, bahwa dimungkinkan setiap Agen Uji berprilaku mewakili 1 kelompok user yang melakukan proses uji tertentu.

Kelemahan dari model ini terletap pada trafik yang dihasilkan homogen, sehingga tidak sesuai dengan kondisi peak season yang sesungguhnya. Kelemahan ini disempurnakan pada penelitian ke-2, yang berfokus pada bagaimana membangkitkan kasus uji dengan metode *back propagation*, seperti pada Gambar 2.4.
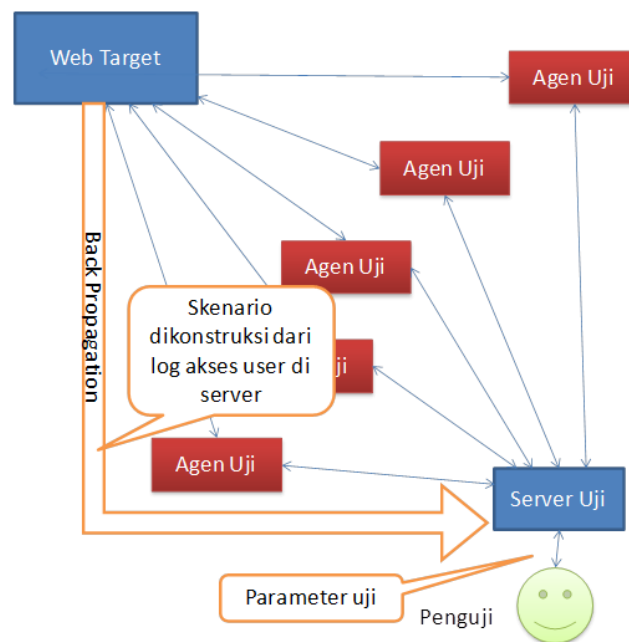

Gambar 2.4. Model Pembangkitan Skenario Uji dengan Metode *Back Propagation*

Dengan menerapkan model pada Gambar 2.4 [25], skenario uji dibangkitkan dari jejak proses user dari Web Target, kemudian diduplikasi dan divariasi sebelum dijadikan kasus uji yang dieksekusi oleh Agen Uji. Model ini menghasilkan trafik yang lebih mewakili kondisi yang sesungguhnya. Model detail dari Gambar 2.4 dapat dilihat pada Gambar 2.5.
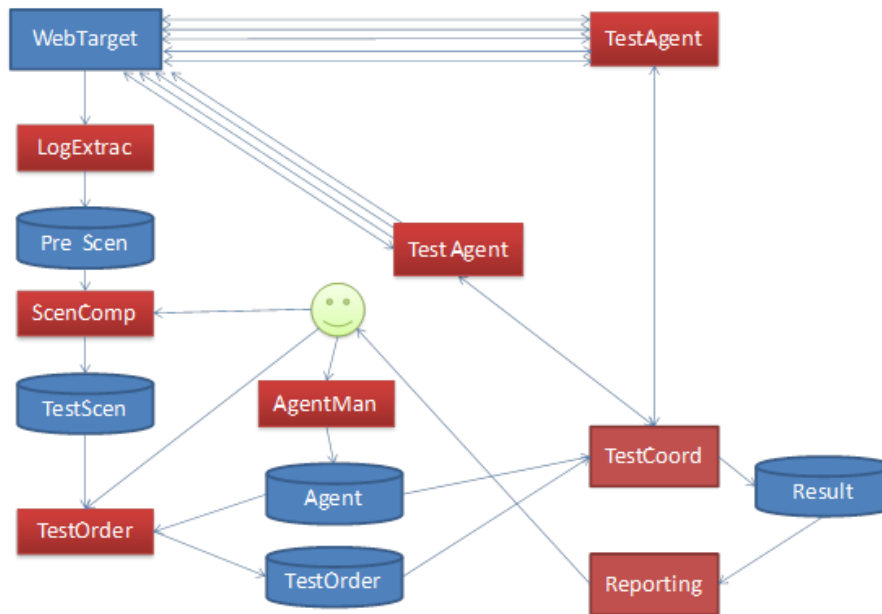
Gambar 2.5. Detail dari Model *Back Propagation*

Pada Gambar 2.5 dimulasi dari proses ekstrasi log menggunakan Mod_Security [26] oleh LogExtrac, kemudian dilanjutkan dengan pembangkitan skenario dan order kepada Agen, sampai dengan menghasilkan laporan dari hasil eksekusi Agen Uji. Konstruksi skenario diadopsi dan diadaptasi dari konsep aplikasi Jmeter [27] Model Gambar 2.5 diimplementasikan pada lingkungan web dan J2SE seperti pada Gambar 2.6.
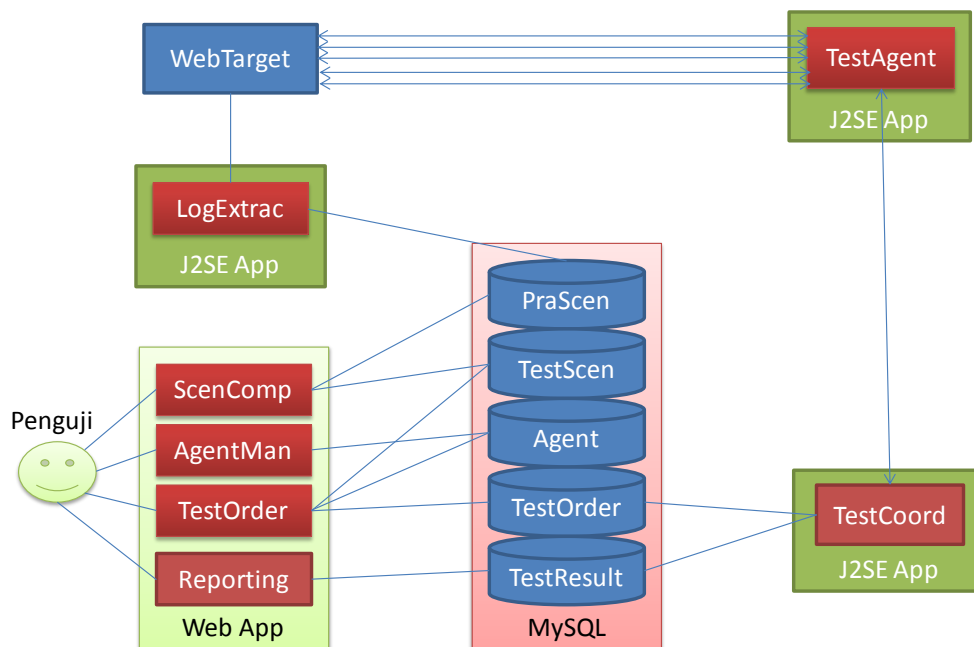

Gambar 2.6. Implementasi Model Back Propagation

Penelitian ke-3 fokus pada aspek fungsional [28], dengan mengembangkan perangkat lunak untuk otomasi uji fungsional. Model yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 2.6.
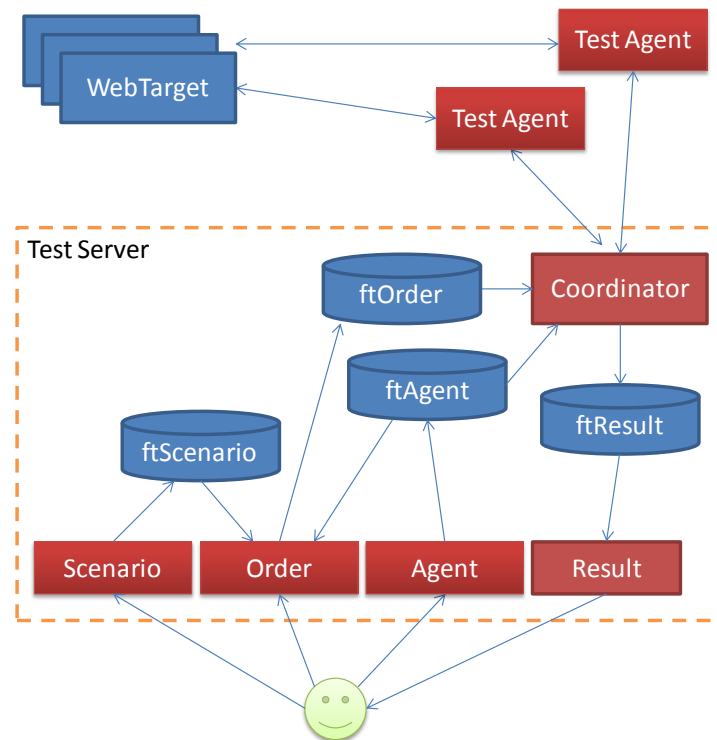
Gambar 2.7. Model Otomasi Uji Fungsional

Pada Gambar 2.7, skenario uji fungsional dibuat oleh tester, kemudian skenario tersebut dipesan untuk dijalankan oleh Agen Uji (Test Agent) secara terkoordinasi dan hasilnya disimpan untuk proses pelaporan dan analisis. Untuk pembuatan skenario telah dikembangkan bahasa yang merepresentasikan proses sekuensial, pengulangan dan repetisi. Model 2.7 diimplementasikan pada lingkungan web dan J2SE dengan basis data MySQL dengan menggunakan library HTMLUnit [29] seperti pada Gambar 2.8.
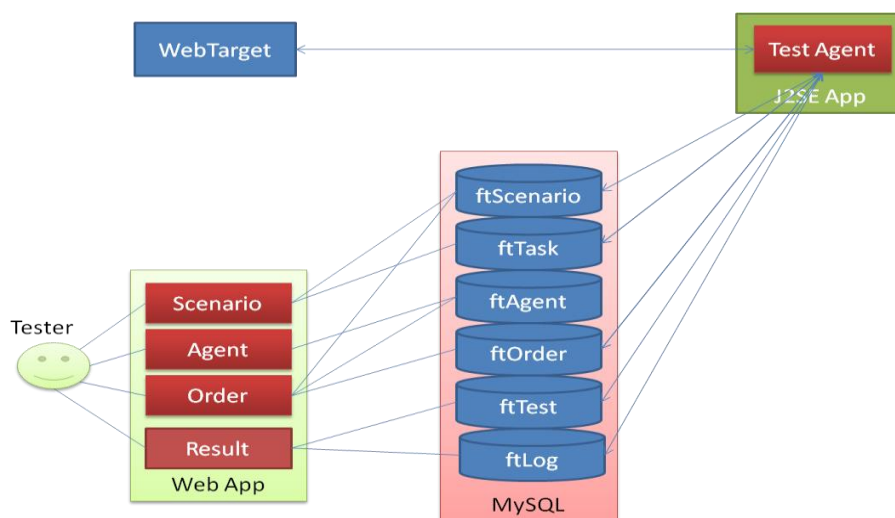
Gambar 2.8 Implementasi Model Otomasi Uji Fungsional

Pada penelitian 4 [30], hasil dari penelitian 2 dan 3 dikemas menjadi satu lingkungan uji terintegrasi dengan menambahkan fungsi pengelolaan. Model pada penelitian 4 dapat dilihat pada Gambar 2.9.
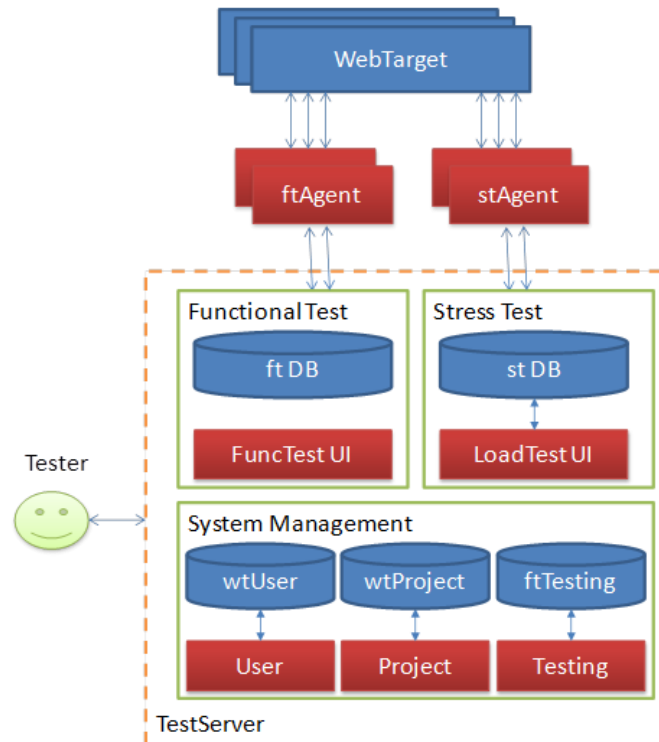
Gambar 2.9. Model IWTE

Pada Gambar 2.9, uji fungsional ditangani oleh FunctionalTest dari penelitian 3, sedangkan uji beban/ kapasitas/ performansi ditangani oleh StressTest dari penelitian 2. Sedangkan aspek manajemen ditangai oleh System Management, yang menangani: user, project dan testing. Model ini mendukung lingkungan multi user, multi project dan multi testing. Model dan fitur ini juga telah dibandingkan dengan software sejenis seperti Selenium [31] dan Watir [32]. Model pada Gambar 2.9 diimplementasikan seperti pada Gambar 2.10.
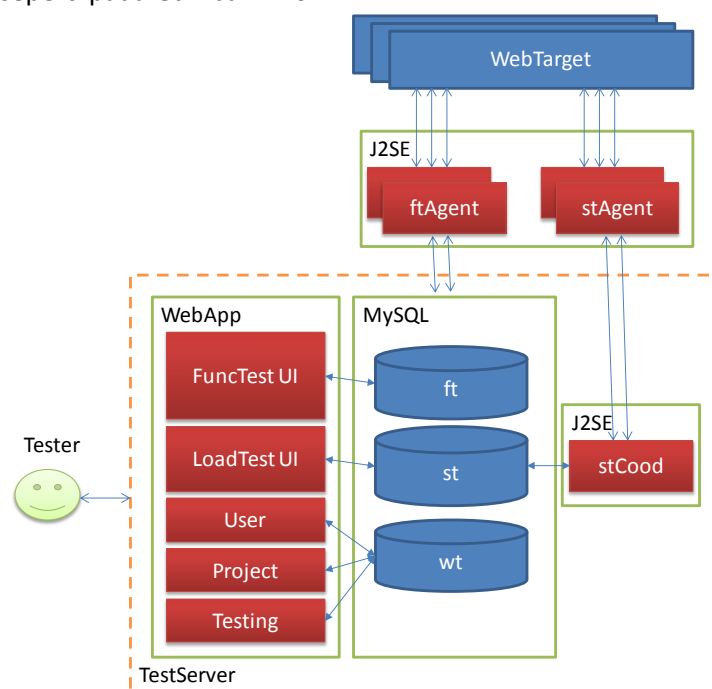
Gambar 2.10 Implementasi Model IWTE

IWTE dapat diakses melalui situs http://webtest.unpar.ac.id. Beberapa tampilan penting yang menunjukkan modul uji fungsional dapat dilihat pada gambar 2.11 sampai dengan 2.14.

Gambar 2.11 Halaman Awal IWTE

Pada Gambar 2.11, tampak halaman awal dari IWTE pada situs http://webtest.unpar.ac.id. Pada halaman ini dapat dilakukan login sesuai dengan pendaftaran yang telah dilakukan.

Gambar 2.12 Halaman Proyek

Pada Gambar 2.12, ditampilkan halaman pengelolaan proyek. IWTE dapat menangani beberapa proyek dan mendukung fitur uji fungsional dan uji beban (*stress test*).

Gambar 2.13 Halaman Uji Fungsional

Pada Gambar 2.13 dapat dilihat halaman uji fungsional. Pada modul ini dapat dibuat skenario untuk melakukan uji fungsional suatu situs web.

Gambar 2.14 Halaman Uji Beban (*Stress Test*)

Halaman uji beban dapat dilihat pada Gambar 2.14. Hasil pengujian beban dapat dilihat pada Gambar 2.15.

Gambar 2.15 Hasil Uji Beban

Dengan demikian IWTE dapat dilakukan untuk melakukan uji fungsional dan uji beban aplikasi berbasis web sekala besar (*web* enterprise).

BAB III. METODE PENELITIAN

Berdasarkan studi pustaka di atas, penelitian ini berkontribusi dalam mengembangkan model audit aplikasi web dalam sekala besar dengan menggunakan *tools* IWTE. Dengan demikian diharapkan tersedia model dan kakas yang memadai dalam melakukan audit aplikasi web enterprise secara berkala dan untuk antisipasi perubahan, khususnya untuk aspek fungsional dan *performance*/ *load*. Berikut dijelaskan tentang metodologi dan jadwal pelaksanaan penelitian.

3.1. Metodologi Penelitian

Pada penelitian ini digunakan metodologi seperti pada Gambar 3.1.
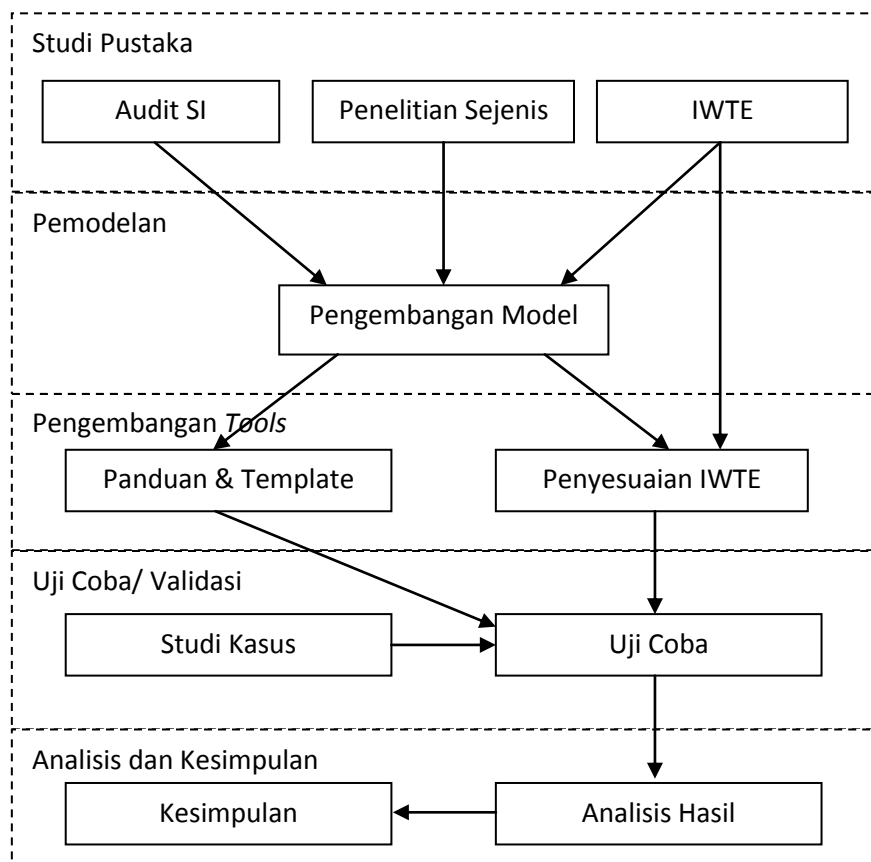
Gambar 3.1. Metodologi Penelitian

Berdasarkan Gambar 3.1, penelitian ini dibagi dalam 5 tahapan, yaitu:

1. **Studi Pustaka**. Studi pustaka tentang standar dan *best practice* audit sistem informasi yang diambil dari sumber *Information System Audit and Control Association* (ISACA), dan teknik pengendalian dari *Control Objective for Information and Technologi related* (CobIT). Selain itu, digali informasi tentang penelitian-penelitian yang telah dilakukan oleh peneliti lain pada bidang sejenis dari makalah-makalah pada jurnal yang relevan. Dengan demikian dapat ditentukan arah kontribusi dari penelitian ini, dan tingkat kebaruan model yang akan dikembangkan. Studi

pustaka juga membahas tentang penelitian IWTE yang digunakan nantinya sebagai dasar untuk menentukan *tools* yang digunakan.

2. **Pemodelan.** Pengembangan model dengan mengadopsi dan mengadaptasi standar dan best practice dari ISACA/COBIT di atas [1][2][3][4][5][6], dengan mempertimbangkan hasil penelitian sejenis terutama [7][8]. Selain itu juga ditetapkan kriteria untuk memverifikasi validitas model (*model evaluation)*. Model ini juga mempertimbangkan visibilitas *tools* yang akan digunakan (IWTE) [30].

3. **Pengembangan *Tools*.** Pada tahap ini, model yang telah dihasilkan diimplementasikan menjadi panduan, template dan alat untuk melakukan audit. Dalam hal ini aspek kepatuhan (*compliance*) menggunakan ceklist, sedangkan aspek substantif menggunakan alat uji IWTE. Pada tahap ini juga dilakukan penyesuaian IWTE ke dalam format audit sesuai dengan model yang dikembangkan**.** Penyesuaian dilakukan dengan menambahkan ceklist kepatuhan dan korelasi antara uji kepatuhan dan uji substantif untuk mengkonstruksi kesimpulan.

4. **Uji coba/ Validasi.** Model yang diimlementasikan dalam bentuk panduan dan alat uji IWTE diujicobakan pada kasus web yang ada di Unpar. Dalam hal ini akan dipilih https://studentportal.unpar.ac.id. Dari hasil uji ini akan dilakukan evaluasi model secara keseluruhan atas pemenuhan terhadap kriteria validitas.

5. **Analisis dan Kesimpulan.** Pada tahapan ini dilakukan analisis hasil uji coba sehingga menghasilkan suatu kesimpulan apakah model yang telah dikembangkan, berikut *tools*-nya berhasil memenuhi kriteria validitasnya.

6. **Desiminasi dan Publikasi.** Pada tahap ini hasil kajian dan uji coba dipublikasikan pada seminar internasional yang relevan. Hasil dari desiminasi digunakan untuk menyempurnakan hasil penelitian. Hasil penelitian yang telah disempurnakan dipublikasikan pada jurnal nasional terakreditasi yang relevan.

3.2. Jadwal Pelaksanaan

Penelitian ini akan dilaksanakan dalam waktu 10 bulan, dengan jadwal pelaksanaan seperti pada tabel 3.1.

Tabel 3.1. Jadwal Pelaksanaan

No	Uraian	Bulan									
		03	04	05	06	07	08	09	10	11	12
1	Studi pustaka	X	X								
2	Pengembangan model		X	X							
3	Pengembangan panduan dan template			X	X						
4	Penyesuaian IWTE				X	X	X				
5	Uji coba/ Validasi						X	X	X	X	X
6	Laporan monitoring							X			
7	Seminar desiminasi							X	X	X	X
8	Penyempurnaan										X
9	Laporan akhir										X

Pada tabel 3.1 dapat dilihat, selain jadwal sesuai dengan kegiatan dalam metodologi, juga ada laporan monitoring dan laporan akhir sesuai dengan ketentuan pelaksanaan penelitian dari LPPM.

BAB IV. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan tentang hasil-hasil penelitian, yang mencakup: tinjauan famework ISACA dalam audit aplikasi web sekala besar (4.1), model dan panduan audit aplikasi web sekala besar (4.2), *tools* untuk menjalankan proses audit (4.3), serta hasil pengujian pada studi kasus (4.4), serta pembahasan (4.5).

4.1. Tinjauan ISACA dalam Audit Aplikasi Web Sekala Besar

Pada [1] ISACA menetapkan *Code of Professional Ethics.* Dari 7 kode etik tersebut, pada point no. 2, menyatakan bahwa seorang auditor harus melaksanakan tugasnya dengan objektif, due diligence dan menjaga profesionalitas sesuai dengan standar profesi. Oleh karena itu, semua auditor bersertifikat CISA diwajibkan memenuhi standar profesi, yaitu *Standards, Guidelines and Tools and Techniques* ini. Pada [1] juga dijelaskan bahwa *standards* menetapkan persyaratan-persyaratan wajib (*mandatory requirements*), sedangkan *guidelines* menyediakan panduan-panduan bagaimana menerapkan standar tersebut. *Tools and techniques* menyediakan informasi bagaimana memenuhi standar pada saat melakukan pekerjaan audit dan penjaminan. *Tools* and techniques lebih lanjut mengacu pada *Information Technology Assurance Framework* (ITAF) [33].

Dari sisi standar, audit terhadap aplikasi web sekala besar berasosiasi dengan standar S6 Performance of Audit Work, S7 Reporting, S14 Audit Evidence. Sedangkan dari sisi panduan, yang berasosiasi adalah G2 Audit Evidence Requirement, G3 Use of Computer Assisted Audit Techniques (CAATs), G8 Audit Documentation, G10 Audit Sampling dan G20 Reporting.

S6 Performance of Audit Work, menyatakan bahwa untuk melaksanakan audit, maka ada 3 persyaratan:
1. *Supervision*, staf audit sistem informasi harus disupervisi untuk mendapatkan keyakinan yang memadai bahwa tujuan audit tercapai dengan memenuhi standar profesional.
2. *Evidence,* selama melaksanakan audit, auditor harus memperoleh bukti yang cukup, dapat diandalkan dan relevan untuk mencapai tujuan audit. Temuan audit dan kesimpulan harus didukung oleh analisis yang tepat dan interpretasi bukti tersebut.
3. *Documentation,* proses audit harus didokumentasikan, menggambarkan pekerjaan audit yang dilakukan dan bukti audit yang mendukung temuan dan kesimpulan auditor

Dalam melaksanakan pekerjaan audit, terdapat panduan penggunaan CAATs (G3) yang menyatakan bahwa CAATs dapat digunakan untuk melaksanakan prosedur audit seperti:
1. Pengujian rincian transaksi dan saldo.
2. Prosedur review analitik.
3. Tes kepatuhan pengendalian umum.
4. Tes kepatuhan pengendalian aplikasi.
5. Pengujian penetrasi (penetration testing).

Dalam konteks CAATs ini IWTE dapat digunakan dalam melakukan tes kepatuhan pengendalian aplikasi dan pengujian penetrasi.

Hal-hal yang perlu dilaporkan dijelaskan lebih lanjut di S7 Reporting, yang panduannya lebih jauh ada pada G20 Reporting. S14 Audit Evidence mensyaratkan auditor untuk:

1. Memperoleh bukti yang cukup dan tepat untuk menarik kesimpulan yang masuk akal yang menjadi dasar hasil audit. Hal ini mencakup prosedur yang dilakukan, hasilnya, dokumen-dokumen sumber (baik dalam format elektronik maupun kertas), catatan-catatan dan informasi lain yang menguatkan untuk mendukung audit. Selain itu juga temuan dan hasil, serta dapat menunjukkan bahwa hal tersebut sesuai dengan hukum, peraturan dan kebijakan yang berlaku pada suatu organisasi yang diaudit (*auditee*).

2. Mengevaluasi kecukupan bukti audit yang diperoleh selama audit. Evaluasi bukti-bukti dapat dilakukan melalui proses: inspeksi, observasi, penyelidikan dan konfirmasi, reperformance, perhitungan, prosedur analisis, dan metode lain yang berlaku umum.

Lebih lanjut tentang bukti audit dijelaskan dalam panduan G2 *Audit Evidence Requirement* dan G8 *Audit Documentation*.

Pembahasan *Tools and Techniques* pada ITAF terorganisasi seperti pada gambar 4.1 [33].
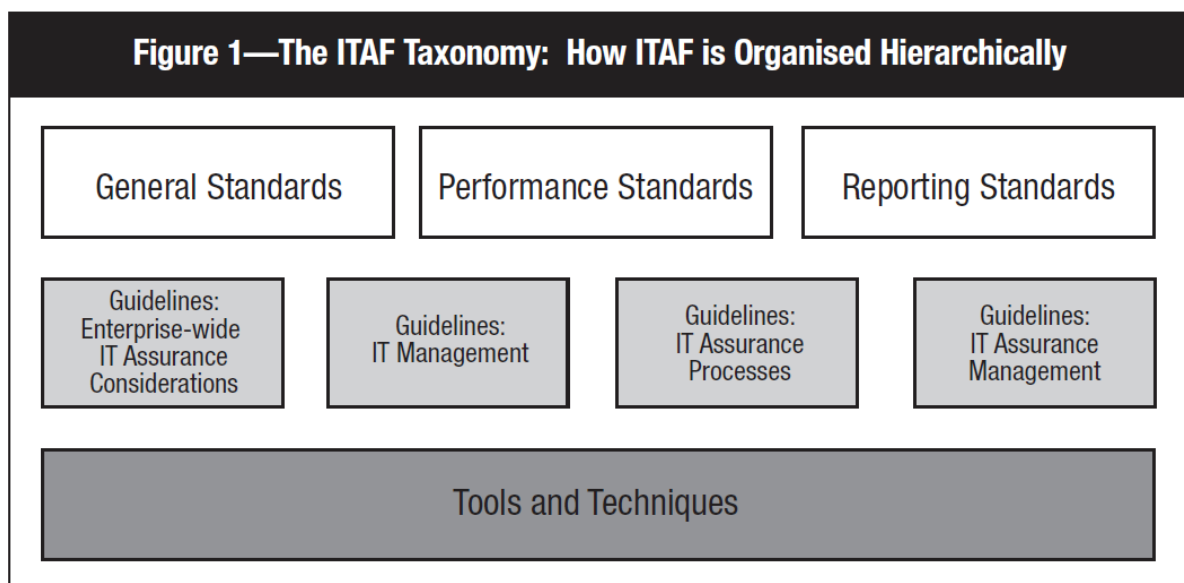
Gambar 4.1. Taxonomy ITAF

Dari gambar 4.1, yang terkait dengan audit aplikasi web skala besar adalah *IT Assurance Processes*, khususnya pada kelompok bagian 3600 IT Audit and Assurance Process bagian 3650 *Auditing Application Controls*. Pada bagian ini dijelaskan proses untuk:

1. Bagian 3653*, auditing traditional application control*, yang biasanya menggunakan pendekatan terstruktur seperti (1) SDLC (System Development Live Cytle). Selain itu juga mencakup *(2) input autorisation, (3) batch control and balancing, (4) input and process editing, (5) rejecting/ suspendese of transactions, (6) batch integrity in online or database system, (7) processing procedures and controls, (8) output control, (9) applicaiton access, (10) log management, (11) end-user computing application,* dan *(12) business intelligence.*

2. Bagian 3655, auditing ERP systems.

3. Bagian 3657, auditing alternative software development strategy, seperti: portal, web service, service oriented software, UML dan strategi yang lain.

4. Bagian 3660 Audit Spesific Requirements, yang dibagi ke dalam kriteris khusus terkait pemerintah, dan kriteria khusus terkait dengan industri.

5. Bagian 3670 Audit With Computer-assisted Audit Techniques (CAATs). Sehubungan dengan penggunaan IWTE, pada ITAF terkait dengan klausul ini, khususnya pada point (1) untuk keperluan System Testing.

Selain itu, pada bagian 3610 Using COBIT in IT Assurance Process, dijelaskan bahwa control practice pada COBIT dapat diadopsi untuk melakukan audit, terutama berkaitan dengan batasan-batasan cakupan dan situasi yang bersesuaian.

Panduan rinci tentang bagaimana mengaudit program aplikasi secara umum ada pada *Generic Application Audit/ Assurance Program* [6]. Pada panduan ini ada 8 program untuk melaksanakan audit aplikasi, yaitu 2 program manajemen:

1. *Planning and Scoping the Audit*
2. *Planning the Application Audit*

Dan 6 program untuk melaksanakan audit:

1. *Source Data Preparation and Authorization*
2. *Source Data Collection and Entry*
3. *Accuracy, Completeness and Authenticity Checks*
4. *Processing Integrity and Validity*
5. *Output Review, Reconciliation and Error Handling*
6. *Transaction Authentication and Integrity*

Detail dari panduan di atas dapat dilihat pada **Lampiran 1.** Selain itu juga dijelaskan tentang *maturity assesment matrix* dari sebuah aplikasi berdasarkan 6 program audit di atas yang diasosiasikan dengan *control practice* di COBIT (**Lampiran 2**) [2][6].

Pada COBIT [2] juga digambarkan bagaimana interaksi antara business dengan aplikasi dalam konteks IT (Gambar 4.2).

Gambar 4.2 Batasan Pengendalian *Business, IT* dan *Application*

Maturity level secara umum didefinisikan dalam COBIT [2] seperti pada gambar 4.3. Definisi masing-masing level kematangan dijelaskan secara umum pada gambar 4.4.

Figure 12—Graphic Representation of Maturity Models

Non-existent | Initial/ Ad Hoc | Repeatable but Intuitive | Defined Process | Managed and Measurable | Optimised
0 | 1 | 2 | 3 | 4 | 5

LEGEND FOR SYMBOLS USED
- Enterprise current status
- Industry average
- Enterprise target

LEGEND FOR RANKINGS USED
0—Management processes are not applied at all.
1—Processes are *ad hoc* and disorganised.
2—Processes follow a regular pattern.
3—Processes are documented and communicated.
4—Processes are monitored and measured.
5—Good practices are followed and automated.

Gambar 4.3 Representasi Grafis dari Model Kematangan

Figure 13—Generic Maturity Model

0 Non-existent—Complete lack of any recognisable processes. The enterprise has not even recognised that there is an issue to be addressed.

1 Initial/Ad Hoc—There is evidence that the enterprise has recognised that the issues exist and need to be addressed. There are, however, no standardised processes; instead, there are *ad hoc* approaches that tend to be applied on an individual or case-by-case basis. The overall approach to management is disorganised.

2 Repeatable but Intuitive—Processes have developed to the stage where similar procedures are followed by different people undertaking the same task. There is no formal training or communication of standard procedures, and responsibility is left to the individual. There is a high degree of reliance on the knowledge of individuals and, therefore, errors are likely.

3 Defined Process—Procedures have been standardised and documented, and communicated through training. It is mandated that these processes should be followed; however, it is unlikely that deviations will be detected. The procedures themselves are not sophisticated but are the formalisation of existing practices.

4 Managed and Measurable—Management monitors and measures compliance with procedures and takes action where processes appear not to be working effectively. Processes are under constant improvement and provide good practice. Automation and tools are used in a limited or fragmented way.

5 Optimised—Processes have been refined to a level of good practice, based on the results of continuous improvement and maturity modelling with other enterprises. IT is used in an integrated way to automate the workflow, providing tools to improve quality and effectiveness, making the enterprise quick to adapt.

Gambar 4.4. Definisi Umum Level Kematangan

4.2. Usulan Model dan Panduan Audit

Berdasarkan ITAF dan Cobit, diusulkan model audit Aplikasi Web Sekala Besar seperti pada Gambar 4.5. Pada Gambar 4.5 tersebut dijelaskan apa yang menjadi input dan output dari proses audit.

Sebuah aplikasi web sekala besar dikembangkan atas dasar persyaratan/ kebutuhan fungsional dan non fungsional. Peryaratan fungsional diwujudkan dalam bentuk *functional requirement,* fokus pada apa (**what**) yang dapat dijalankan oleh produk tersebut [ISO/IEC9126]*.* Spesifikasi ini dibuat atas dasar proses bisnis yang diotomasi pada aplikasi tersebut. Sebuah spesifikasi kebutuhan fungsional harus mencakup: prosesnya apa, pelakunya siapa (user) dan data apa yang dimasukkan, serta informasi apa yang dihasilkan, bagaimana cara mempresentasikannya.
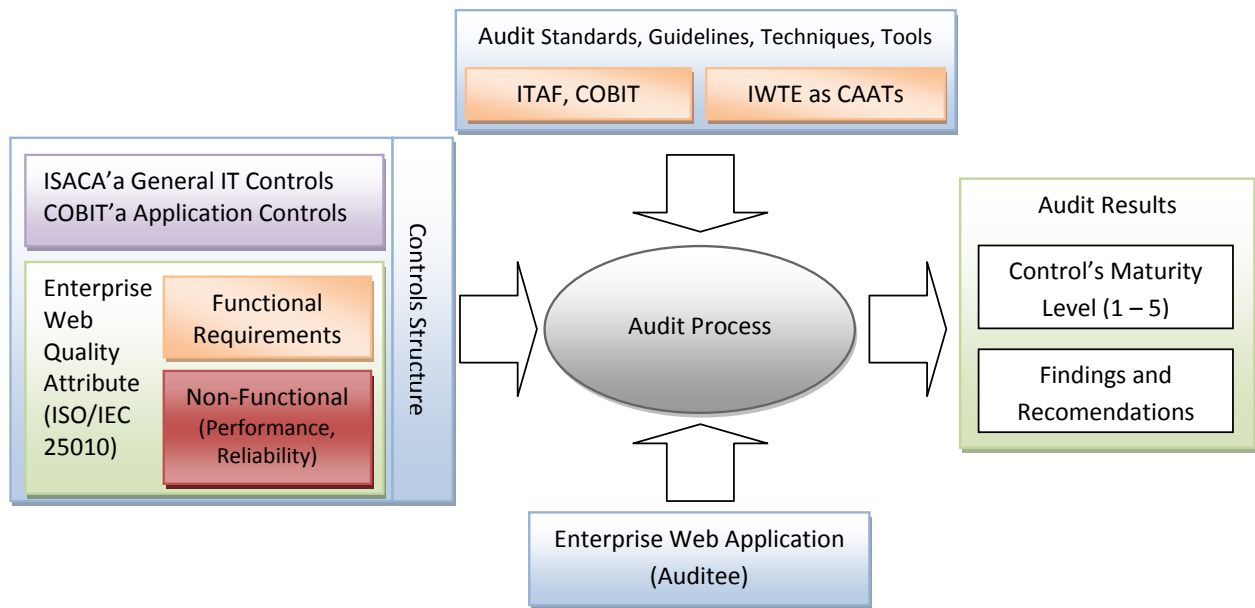
Gambar 4.5. Usulan Model Audit Aplikasi Web Sekala Besar

Persyaratan-persyaratan non fungsional (*non functional requirements*), fokus pada sejauh mana (**how well**) fungsi dari perangkat lunak berjalan [ISO/IEC9126], mencakup hal-hal sebagai berikut:

1. **Performansi [IEEE 610.12:1990]**, termasuk diantaranya: *concurent user* (berdasarkan perencanaan kapasitas), *response time* (berdasarkan kecukupan waktu operasional), *throghput* (data dan teransaksi).

2. **Reliability [IEEE 610.12:1990]** dan **availability** (berdasarkan SLA yang disyaratkan), misalnya: apakah suatu aplikasi web sekala besar harus berjalan/ tersedia 24 jam sehari 7 hari seminggu?
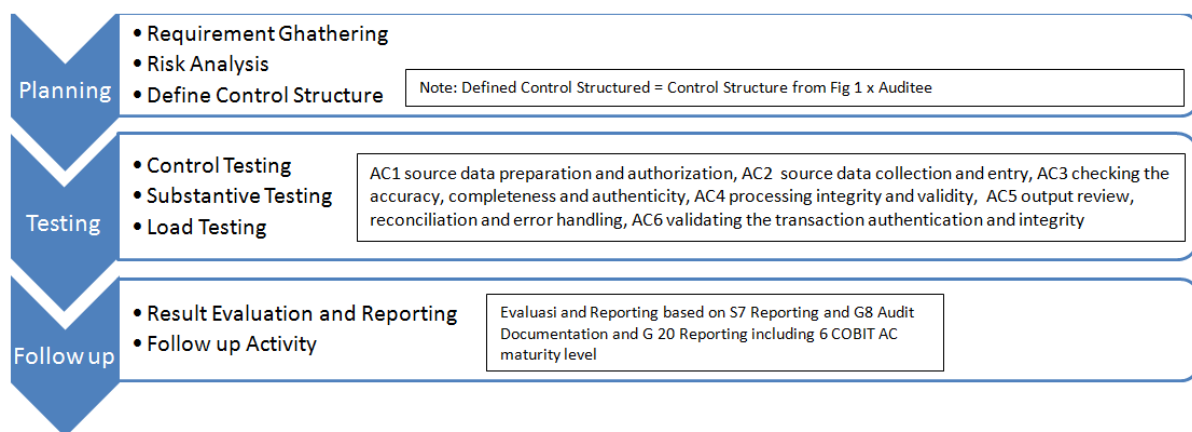
3. **Maintenability**, terkait dengan tingkat kemudahan dalam pemeliharaan, termasuk dalam hal ini untuk mengantisipasi perubahan atau penambahan fitur akibat dari adanya perubahan atau penambahan kebutuhan (persyaratan).

4. **Usability [ISO/IEC 9126-1 (2000)]**, sejauh mana suatu aplikasi web dengan mudah digunakan oleh para penggunanya. Diantaranya termasuk, manual yang mudah dipelajari, skenario interaksi yang intuitif dan ada unsur daya tarik yang menyebabkan pengguna tertarik untuk menggunakannya.

5. **Security [BS7925-1:1998]**, sejauh mana aplikasi memenuhi aspek keamanan. Dalam aplikasi web, keamanan mencakup keamanan akses, dimana hanya pihak yang boleh yang dapat melakukan akses, sedangkan harus dapat dipastikan bahwa pihak yang tidak boleh tidak dapat mengaksesnya. Untuk keperluan ini perlu mekanisme otentifikasi dan otorisasi yang sesuai. Juga berkaitan dengan perlindungan terhadap protokol komunikasi sehingga memberikan jaminan tidak ada pihak yang tidak berwenang dapat mengakses data/ traksaksi yang dikomunikasikan.

Audit didasarkan atas standar, panduan, teknik yang telah diuraikan pada bagian 4.1. Untuk kakas audit diuraikan lebih lanjut pada bagian 4.3.

Audit dilakukan terhadap auditee (aplikasi web sekala besar). Sedangkan output dari proses audit adalah tingkat kematangan berdasarkan matrix kematangan yang terdiri atas 6 aspek kematangan aplikasi. Sedangkan untuk hasil kesesuaian antara persyaratan baik fungsional dan non fungsional itu terdapat pada bagian temuan dan rekomendasi.

Adapun proses audit digambarkan seperti pada Gambar 4.6. Pada Gambar 4.6 dapat dilihat bahwa ada 3 tahapan proses audit, yaitu:

1. Tahap Perencanaan, mancakup: identifikasi persyaratan, analisis resiko dan penentuan cakupan.
2. Tahap Pengumpulan dan Evaluasi Fakta, mencakup: proses pengumpulan fakta/ bukti/ evidence, dan menguji kehandalan dan kecukupannya, serta menguji substansinya.
3. Tahap Evaluasi Hasil dan Penyusunan Rekomendasi, mencakup: evaluasi hasil, pelaporan dan review tindaklanjut dengan auditee.

Planning
- Requirement Ghathering
- Risk Analysis
- Define Control Structure

Note: Defined Control Structured = Control Structure from Fig 1 x Auditee

Testing
- Control Testing
- Substantive Testing
- Load Testing

AC1 source data preparation and authorization, AC2 source data collection and entry, AC3 checking the accuracy, completeness and authenticity, AC4 processing integrity and validity, AC5 output review, reconciliation and error handling, AC6 validating the transaction authentication and integrity

Follow up
- Result Evaluation and Reporting
- Follow up Activity

Evaluasi and Reporting based on S7 Reporting and G8 Audit Documentation and G 20 Reporting including 6 COBIT AC maturity level

Gambar 4.6. Langkah Proses Audit Aplikasi Web Sekala Besar

Berikut adalah penjelasan dari masing-masing tahapan pada Gambar 4.6:

Tahap 1 Perencanaan, dilakukan survei pendahuluan berupa *document gathering* untuk mengumpulkan persyaratan-persyaratan (*requirement*) termasuk di dalamnya aspek fungsional dan non fungsional. Aspek non-fungsional yang relevan dengan aplikasi web sekala besar adalah aspek: performansi (kapasitas, konkurensi, response time) dan reliabilitas (ada berapa sub sistem yang berinteraksi). Atas dasar persyarata-persyaratan tersebut dianalisa tingkat resiko atas pemenuhan masing-masing persyaratan menggunakan pendekatan BIA (*business impact analysis*). Sebagai contoh, jika persyaratan kapasitas tidak terpenuhi, akan menimbulkan kerugian berupa denda, atau tertundanya jadwal operasional. Atas dasar analisis resiko tersebut diusulkan proses pendendalian (*control process*) yang relevan dengan resiko tersebut. Untuk setiap resiko diperingkat atas dasar materialitas dari resiko tersebut. Materialitas dapat menggunakan sekala kualitatif (tinggi, sedang, rendah) atau jika memungkinkan menggunakan sekala kuantitatif berdasarkan ukuran tingkat kerugian dan probabilitas kejadian. Lebih jauh contoh hasil BIA ini dapat dilihat pada Tabel 4.1

Tabel 4.1. *Business Impact Analysis*

No	Requirement	Risk		Control
		Description	Materiality	
1	*Functional requirement* seperti: login, frs, daftar nilai.	Tidak berfungsi baik	Tinggi	Dilakukan pengujian fungsional untuk mengatasi resiko kinerja tidak sesuai
2	*Performance requirement*, seperti: dapat melayani 10.000 user dalam waktu 1 hari. Minimal dapat melayani 500 user secara bersamaan	Performansi server tidak diketahui Performansi server tidak sesuai persyaratan minimum	Tinggi Tinggi	Dilakukan uji beban untuk mengetahui performansi sesungguhnya. Kapasitas server ditingkatkan sampai mencapai kebutuhan minimum + 20%
3	*Control requirement*, seperti: prosedur terkait penyiapan data, verfikasi, otorisasi dan keamanan	Prosedur penyiapan data belum ada	Sedang	Dilakukan uji pengendalian menggunakan 6 AC dari Cobit 4.1.

Atas dasar matrix BIA diurutkan berdasarkan *Risk Materiality* dari yang besar ke yang kecil. Atas dasar pertimbangan waktu dan sumber daya yang dialokasikan untuk proses audit, maka dipilih *requirement* yang memiliki tingkat materialitas yang tinggi.

Atas dasar hasil pemeringkatan di atas, maka dapat disusun struktur pengendalian (*control structure*) yang akan menjadi cakupan proses audit. Jadi daftar pengendalian inilah yang akan dijadikan sasaran audit.

Tahap 2 Pengumpulan dan Evaluasi Fakta. Pada tahap ini dikumpulkan bukti-bukti untuk melakukan pengujian kehandalan pengendalian dan kecukupannya (*complience test*) dan jika diperlukan dilakukan pengujian substantif (*substantive test*). Dalam kaitannya dengan aplikasi web sekala besar, maka aspek performansi dan reliabilitas menjadi aspek yang perlu diuji secara substantif selain aspek fungsionalitas. Evaluasi kehandalan dan substansi menjadi hal utama yang dilakukan pada tahap ini.

Control Testing. Pengujian kehandalan pengendalian dan kecukupan dapat menggunakan *maturity assesment* yang mengacu pada 6 AC dari Cobit 4.1, seperti diuraikan pada bagian 4.1 di atas. Langkah untuk melakukan pengujian kematangan keenam pengendalian aplikasi di atas dapat menggunakan panduan seperti pada **Lampiran 2** yang formnya dikembangkan melengkapi IWTE pada modul pengujian pengendalian (*control testing*). Disain form tersebut diuraikan lebih lanjut pada bab 5 laporan ini atas dasar template *control assesment* yang dapat dilihat pada **Lampiran 3**.

Substantive Testing. Pengujian substantive dapat menggunakan teknik uji fungsional yang dalam penelitian ini menggunakan bantuan IWTE, khususnya modul pengujian fungsional (*functional testing*).

Load Testing. Pengujian beban untuk memenuhi menguji pemenuhan kriteria non fungsional, yaitu: *performance* dan *reliability* juga dapat menggunakan bantuan IWTE, khususnya modul *load testing*. Teknik pengujian yang dilakukan dengan menggenerate kasus uji dengan concurent user tertentu, dan dijalankan secara simultan dalam waktu tertentu. Selain menghasilkan indikator kapasitas yang dapat ditangani, juga menguji apakah dalam rentang waktu tertentu, sistem dapat beroperasi secara terus menerus (reliable).

Tahap 3 Evaluasi Hasil dan Tindaklanjut. Pada tahap ini dilakukan evaluasi akhir dan dituangkan dalam bentuk laporan. Di dalam laporan, selain temuan dan rekomendasi, juga diberikan *level maturity* dari masing-masing pengendalian berdasarkan matrix *maturity level* seperti pada Gambar 4.7.
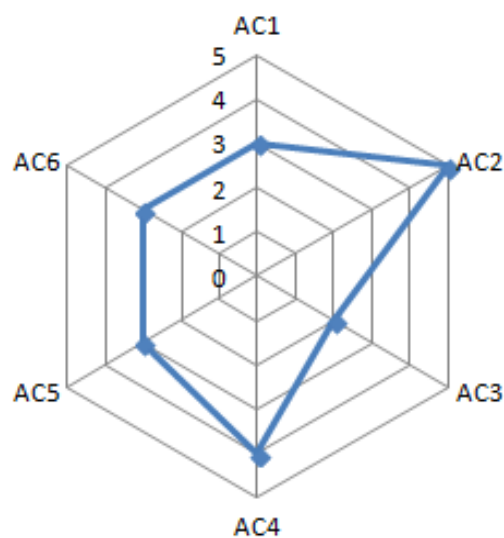
Gambar 4.7. *Spider Diagram* menyatakan *Application Control Maturity Level*

Laporan hasil evaluasi direview bersama dengan *auditee*, sehingga menimbulkan komunikasi dua arah, dan komitmen tindaklanjut dari *auditee*. Tindaklanjut bisa berupa sekedar klarifikasi, tapi juga bisa berupa komitmen perbaikan proses pengendalian dalam rangka *continuous improvement*.

4.3. *Tools* Audit Aplikasi Web Sekala Besar

Untuk melakukan audit aplikasi web sekala besar menggunakan model yang dijelaskan pada bagian 4.2, diperlukan *tools* yang memiliki kemampuan sebagai berikut:
1. Menyediakan fitur uji pengendalian (*control testing*).
2. Menyediakan fitur uji fungsional (*substantive testing).*
3. Menyediakan fitur uji performansi dan reliabilitas (*load testing*).

Sesuai dengan penjelasan pada bagian 2, bahwa aplikasi IWTE telah menyediakan modul untuk melakukan uji fungsional (*substantive*) dan uji beban (*load*). Oleh karena itu, pada penelitian ini ditambahkan dengan modul tambahan untuk uji pengendalian (*control testing*) yang mengimplementasikan 6 AC dari Cobit 4.1. Dengan demikian model IWTE berubah dari Gambar 2.9 menjadi seperti pada Gambar 4.8.
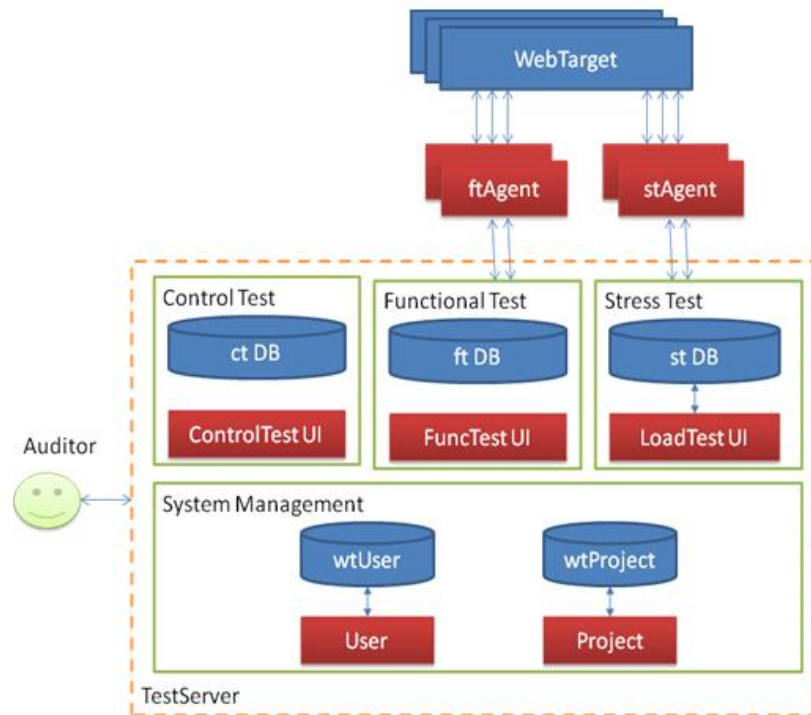
Gambar 4.8. Model *IWTE* untuk *Web Enterprise Audit*

Pada Gambar 4.8 dapat dilihat bahwa IWTE terdidi atas 4 modul, yaitu: modul *System Management* yang bertugas menangani multi user dan multi project. Modul *Control Test* yang bertugas menangani *control assesment*. Sedangkan modul *Functional Test* dan *Stress Test* masing-masing untuk menangani otomasi uji fungsional dan beban dari aplikasi web yang menerapkan prinsip multi agen.

Detail disain dan implementasi dari modul *Control Test* dapat dilihat lebih lanjut pada Gambar 4.9.
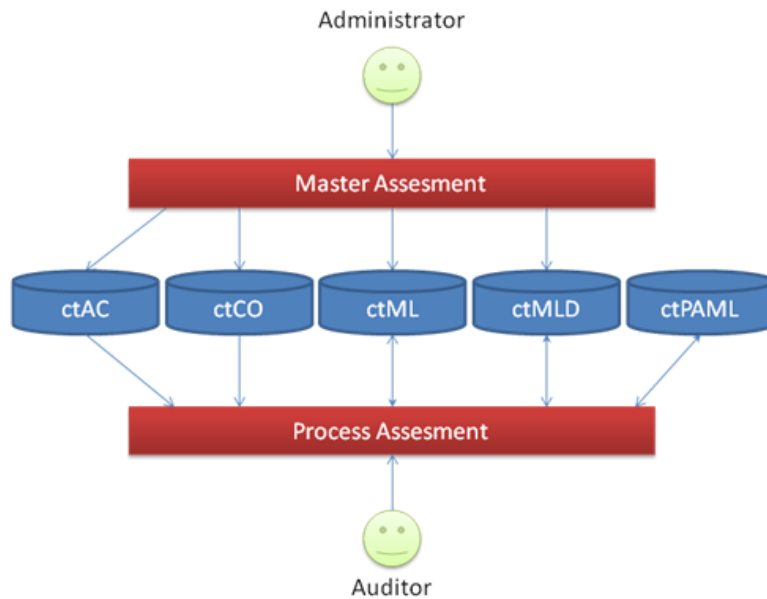
Gambar 4.9. Disain Modul *Control Test* pada IWTE

Pada Gambar 4.9 dapat dilihat bahwa modul *Control Test* terdiri atas 5 entitas yang disimpan sebagai basis data **ct** (*control testing*). Adapun tabel-tabel untuk mengimplementasikan setiap entitas dapat dijelaskan sebagai berikut:

1. *Application Control*, disingkat ctAC, menyimpan 6 pengendalian aplikasi yang didefinisikan berdasarkan Cobit 4.1. Struktur dari tabel tersebut sebagai berikut:

    ctAC(<u>ac_no</u>, statement)

2. *Control Objective*, disingkat ctCO, menyimpan control objective dari masing-masing pengendalian. Struktur dari tabel tersebut sebagai berikut:

    ctCO(<u>ac_no, co_no</u>, objective , fk (ac_no=ctAC.ac_no))

3. *Maturity Level*, disingkat ctML, menyimpan informasi tentang maturity level dari setiap control objective. Struktur tabelnya sebagai berikut:

    ctML(<u>ac_no, level</u>, complience, contribution, value, fk (ac_no=ctAC.ac_no))

4. Maturity Level Description, disingkat ctMLD, menyimpan deskripsi indikator dari level maturity dari suatu control objective. Struktur tabel ctMLD sebagai berikut:

    ctMLD (<u>ac_no, level, no</u>, statement, weight, value, relative_importance, fk ((ac_no,level)=(ctML.ac_no, ctML.level)))

5. Process Assesment Maturity Level, disingkat ctPAML, menyimpan hasil assesment dari suatu proyek. Struktur tabel ctPAML sebagai berikut:

    ctPAML (<u>pa_id,</u> prj_id, ac_no, value, fk (prj_id=wtProject.prj_id))

Contoh isian dari tabel-tabel di atas sebagai berikut:

> ctAC (1, "Persiapan sumber data dan orotisasi"), (3, "Cek Akurasi, Kelengkapan, dan Keaslian")
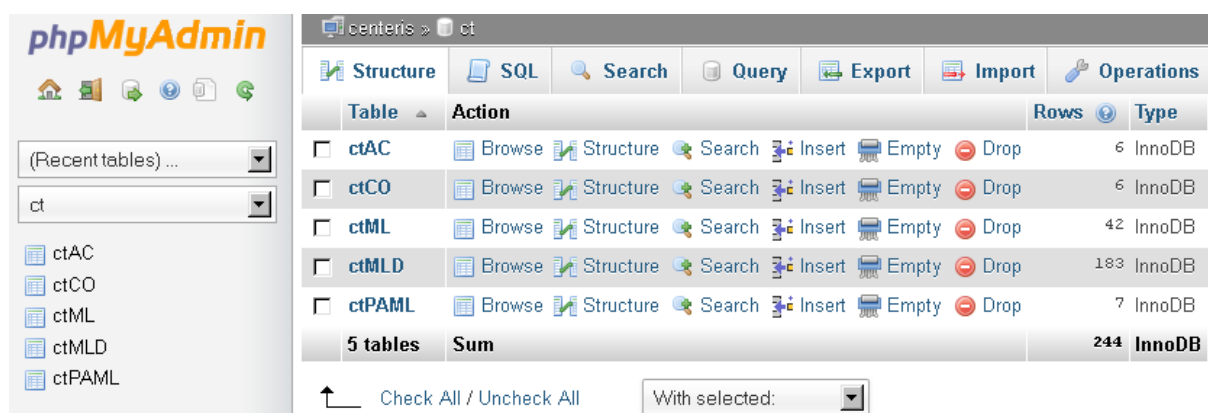>
> ctCO(1, 1, "Pastikan ..."),(1,2,"Meminimalkan ...")
>
> ctML(1,1,1.0,1.0,1.0)
>
> ctMLD(1,1,1,"Kebutuhan akan sumber dokumen telah disadari oleh manajemen",5,0.66,3.3)
>
> ctPAML(1,1,1,2.5)

Basis data ct diimplementasikan pada MySQL Server 5.x. Hasil implementasinya dapat dilihat pada Gambar 4.10.

Gambar 4.10. Implementasi Basis Data ct pada MySQL Server

Adapun proses untuk mengelola *Control Test* ada 2, antara lain:

1. Proses *master assesment* untuk menyimpan ceklist *maturity assesment* untuk 6 AC. Algoritma dari proses ini adalah:
   a. Masukkan judul AC ke tabel ctAC.
   b. Masukkan tujuan setiap AC ke tabel ctCO.
   c. Masukkan level 0-5 dan contribution untuk setiap AC ke tabel ctML.
   d. Masukkan statement dan weight untuk setiap pernyataan kematangan dari setiap AC pada setiap level ke tabel ctMLD.

   Proses penyiapan data master ini dilakukan langsung dengan mengentri ke basis data menggunakan aplikasi PHP MyAdmin.

2. Proses *assesment* untuk suatu aplikasi, mencakup layanan sebagai berikut:
   a. Menampilkan menu *assesment* yang berisi 6 *AC*.
   b. Menampilkan form *assesment* seperti pada form EXCEL (lihat **Lampiran 3**).
   c. Merekap hasil berupa *spider diagram* untuk semua AC.

Proses *assesment* diimplementasikan berupa modul *control assesment* pada IWTE dengan tampilan seperti pada gambar 4.11 dan 4.14.
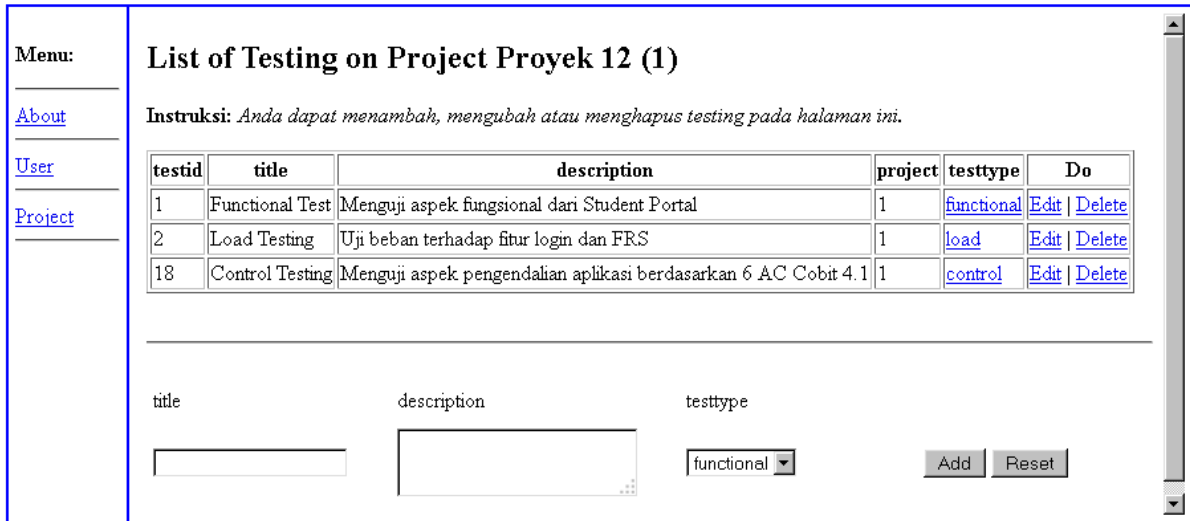
Gambar 4.11. Tampilan Daftar Testing dari Proyek 12 (Student Portal)

Pada Gambar 4.11 dapat dilihat, bahwa Proyek 12 (Student Portal) memiliki 3 jenis test, yaitu: *functional, load* dan *control*. Jika dipilih testid no 18 yang berjenis *control*, maka hasilnya seperti pada Gambar 4.12.
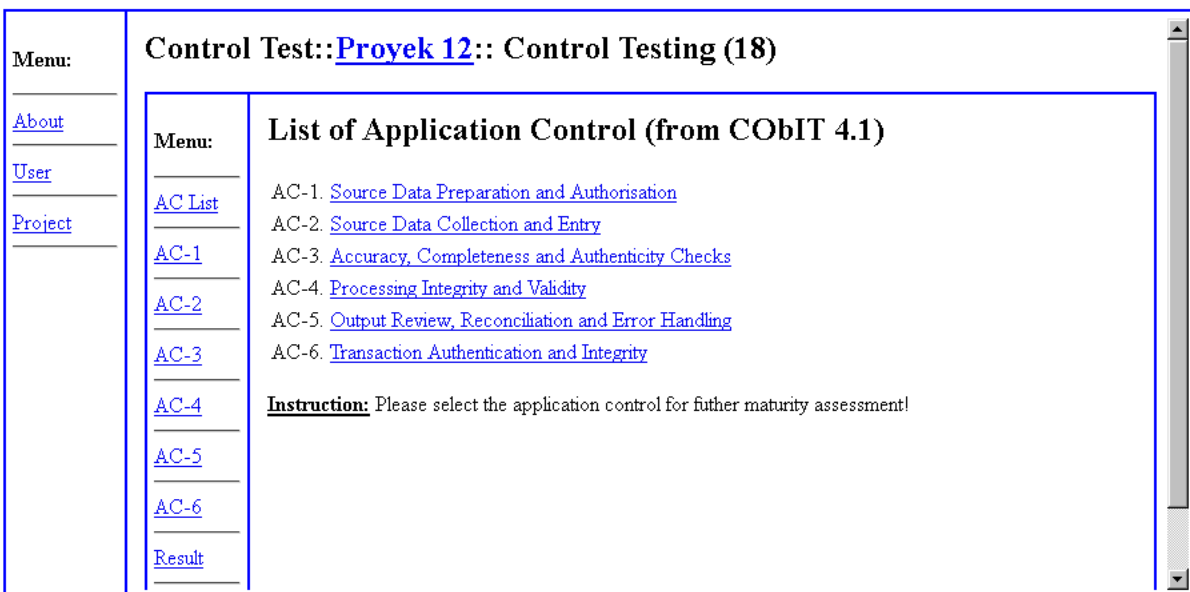
Gambar 4.12. Halaman Utama dari *Control Testing*

Pada Gambar 4.12 dapat dilihat menu dan daftar application control berdasarkan Cobit 4.1. *Assesment* dapat dilakukan dengan memilih salah satu AC melalui menu atau halaman utama. Form untuk *assesment* dapat dilihat pada Gambar 4.13.

Gambar 4.13. Tampilan Proses *Assesment* pada Modul *Control Test*

Pada Gambar 4.13 dapat dilihat ceklist untuk AC 1 Source Data Preparation and Authorisation. Pengisian dilakukan dengan memberikan tanda bagian "*Do you agree*" dengan memililih sesuai dengan kesesuaian kriteria dengan kondisi lapangan dari aplikasi yang diuji. Jika sudah selesai mengisi AC 1 sampai dengan AC6, maka hasil berupa maturity level (spider diagram) dapat dilihat seperti pada Gambar 4.14.
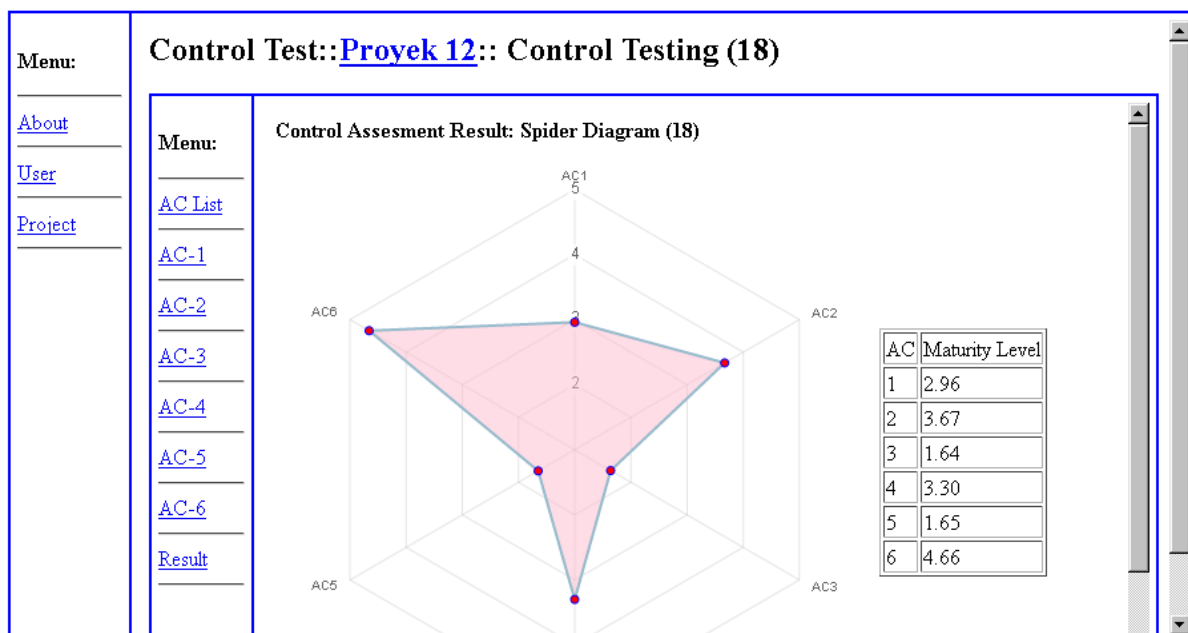
Gambar 4.14. Hasil *Control Test* berupa Spider Diagram

Pada Gambar 4.14 dapat dilihat *maturity level* dari AC1 – AC6 dari Proyek 12 (Student Portal).

5.4. Pengujian Model Pada Kasus Audit Aplikasi Student Portal

Pada bagian ini dijelaskan tentang pengujian model dengan audit kasus Student Portal. Audit dilakukan terhadap emulator Student Portal Unpar yang dapat diakses pada situs http://centeris.unpar.ac.id/sample. Audit tidak dilakukan terhadap Student Portal secara langsung, karena alasan operasional.

Fitur dari Student Portal yang akan dijadikan kasus uji adalah fitur FRS. Proses untuk FRS diawali dengan proses login. Setelah itu dilanjutkan dengan proses pemilihan mata kuliah yang diambil, dan diakhiri dengan proses menampilkan daftar mata kuliah hasil pendaftaran. Detail fitur-fitur dari emulator Student Portal Unpar secara lengkap dapat dilihat pada **Lampiran 4**.

Aplikasi emulator Student Portal Unpar kini terinstal pada mesin server virtual dengan spesifikasi *processor* 4 core 3 GHz, dengan RAM 8 Gbyte dan storage 50 GB. Pada server juga diinstall IWTE dengan alamat akses http://webtest.unpar.ac.id.

5.4.1. Skenario Audit Aplikasi Student Portal

Audit dilakukan dengan menerapkan tahapan yang telah diusulkan pada bagian 4. Audit dilakukan dengan tahapan sebagai berikut:

1. **Tahap Perencanaan**, mancakup: identifikasi persyaratan, analisis resiko dan penentuan cakupan. Pada tahap ini diidentifikasi persyaratan (*requirement*) dari Student Portal, yang dikelompokkan menjadi functional dan non functional. Khusus untuk *control requirement* mengikuti *application control* yang telah didefinisikan. Berdasarkan seluruh persyaratan, dibuat analisis resiko dengan mengisi matrix *business impac analysis*. Atas dasar matriks resiko tersebut, diambil 3 tertinggi.

Tabel 4.2. *Business Impact Analysis Aplikasi Student Portal*

No	Requirement	Risk		Control
		Description	**Materiality**	
1	Dapat melayani mahasiswa yang telah terdaftar di Unpar	Dapat diakses oleh mahasiswa yang tidak teregistrasi	Tinggi	Ada prosedur dan fasilitas identifikasi berupa login menggunana identitas mahasiswa Unpar
2	Dapat melayani fungsi registrasi (FRS) mahasiswa	Tidak berfungsi baik	Tinggi	Dilakukan pengujian fungsional agar berfungsi baik
3	Dapat fungsi penampilan	Tidak berfungsi	Sedang	Dilakukan pengujian fungsional

	jadwal kuliah	baik		agar berfungsi baik
4	Dapat melayani pengecekan tagihan kuliah	Tidak berfungsi baik	Sedang	Dilakukan pengujian fungsional agar berfungsi baik
5	Prosedur terkait penyiapan data, verifikasi, otorisasi dan keamanan untuk mencegah kesalahan data	Prosedur penyiapan data belum ada	Tinggi	Dilakukan uji pengendalian menggunakan 6 AC dari Cobit 4.1.
6	Sistem harus dapat melayani 10 ribu mahasiswa dalam 1 hari atau 500 mahasiswa secara bersama-sama	Kapasitas/ kinerja sistem belum diketahui	Tinggi	Perlu dilakukan uji kapasitas/ kinerja berupa *load test*
		Kapasitas/ kinerja sistem beleum terpenuhi	Tinggi	Perlu dilakukan peningkatan sistem, baik server maupun komponen basis data dan aplikasi serta jaringan.

Berdasarkan BIA pada Tabel 5.1. maka diambil 4 kebutuhan dengan resiko tertinggi, yaitu: no 1, 2, 5 dan 6. Untuk 2 menggunakan uji fungsional, sedangkan untuk 5 menggunakan uji pengendalian serta 6 menggunakan uji beban (*load test*).

2. **Tahap Pengumpulan dan Evaluasi Fakta**, mencakup: proses pengumpulan fakta/ bukti/ *evidence*, dan menguji kehandalan dan kecukupannya, serta menguji substansinya. Pada tahap ini dilakukan *control testing, functional testing* dan *load testing* menggunakan *tools* IWTE. Untuk ini, dibuat proyek baru StudentPortalAudit (Proyek 12), kemudian dilakukan langkah sebagai berikut:

   a. *Control Testing*: dengan mengisi form *assesment*, kemudian ditampilkan *spider diagram* berdasarkan hasil pengisian. Untuk *assesment* sesungguhnya dapat dilakukan bersama dengan perwakilan user dan pengelola sistem. Namun untuk kasus ini, diisi sesuai dengan pengamatan dari peneliti saja.

   b. *Substantive Testing*: dilakukan dengan membuat skenario testing fungsional pada proyek StudentPortal. Audit terutama untuk fungsi dari login sampai dengan melakukan FRS. Setelah itu, skenario diorder terhadap Agen1 untuk dijalankan. Hasilnya dilihat pada menu hasil untuk dianalisis. Berdasarkan analisis ini dibuat daftar temuan.

   c. *Lost Testing*: dilakukan dengan membuat skenario testing beban pada modul *stress test* menggunakan skenario fungsional login dan FRS dengan sample user sebanyak 1.600.

Eksekusi dilakukan di Lab Komputer FTIS menggunakan 40 komputer untuk mengemulasi 40 x 4 x 10 = 1.600 user. Pada saat eksekusi oleh agen-agen yang di-*deploy* komputer lab, performansi server diamati terutama banyaknya *concurrent user* yang aktif dilayani. Setelah proses pengujian selesai, maka log hasil uji dianalisa untuk mengetahui hasilnya. Untuk uji reliabilitas, maka testing beban dilakukan secara terus menerus dalam 2 jam.

3. **Tahap Evaluasi Hasil dan Penyusunan Rekomendasi**, mencakup: evaluasi hasil, pelaporan dan review tindaklanjut dengan *auditee*. Pada tahapan ini semua informasi hasil pengujian di atas dicatat, dan dikatagorikan sebagai capaian, dan temuan. Selain itu juga dilampirkan *maturity level* dari proses pengendalian. Review dengan *auditee* tidak dilakukan pada tahap ini, karena sifatnya internal.

4.4.2. Hasil Audit Aplikasi Student Portal

Berikut adalah hasil dari pengujian model berdasarkan audit kasus Student Portal. Dalam hal ini akan diuraikan hasil *control testing*, *functional testing* dan *load testing*.

1. *Control Testing*

Setelah dilakukan pengisian ceklist *application control assesment* dapat disimpulkan hasil seperti pada Gambar 4.15.
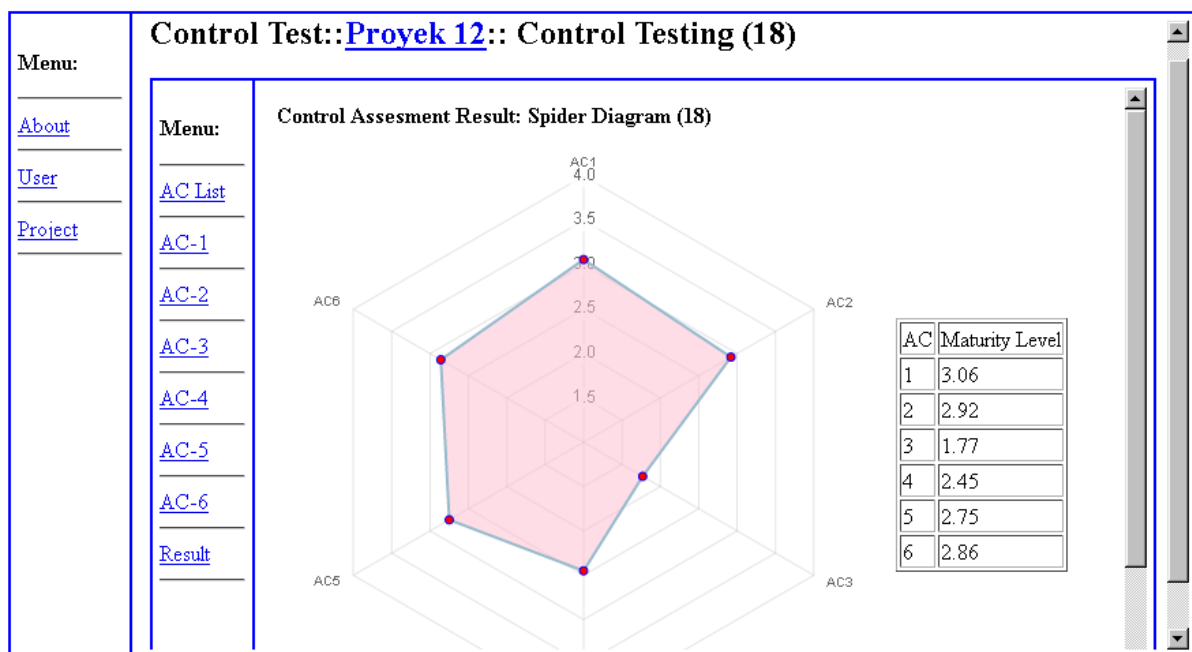
Gambar 4.15. Hasil *Control Testing* Proyek 12 (Student Portal)

Pada Gambar 4.15 tampak bahwa aspek AC 1 *Source Data Preparation and Authorisation* mendapat nilai 3.06 berarti pada tingkatan *defined.* Artinya aspek ini telah terdokumentasi dan

terkomunikasikan. Sedangkan aspek AC 2 *Source Data Collection and Entry,* AC4 Processing Integrity and Validity, AC 5 *Output Review, Reconciliation and Error Handling* dan AC6 *Transaction Authentication and Integrity* hampir seluruhnya telah terdokumentasi dan terkomunikasikan. Hal ini masih perlu ditingkatkan minimal pada level 3. AC 3 *Accuracy, Completeness and Authenticity Checks* bernilai 1.77 menyatakan masih pada tahapan *ad-hoc*. Hal ini perlu ditingkatkan dengan membuat dokumen standar dan mensosialisasikan serta memonitor hasilnya.

2. *Functional Testing*

Pengujian fungsional dilakukan untuk proses login dan FRS sesuai dengan analisis resiko pada Tabel 4.2. Pada Gambar 4.16 dapat dilihat skenario uji fungsional yang digunakan.

Gambar 4.16. Skenario *Functional Test* Proyek 12 (Student Portal)

Pada Gambar 4.16 tampak tasks dari skenario LoginFRSSP. Hasil dari pengujian ini dapat dilihat pada Gambar 4.17.

Gambar 4.17. Hasil *Functional Test* Proyek 12 (Student Portal)

Detail log hasil uji dapat dilihat pada Gambar 4.18. Pada gambar tersebut tampak *task* demi *task* telah diuji beserta hasilnya.

Gambar 4.18. Log Detail Hasil *Functional Test* Proyek 12 (Student Portal)

3. *Load Testing*

Load testing atau sering juga disebut *stress testing* terhadap Student Portal dilakukan dengan mengerahkan 40 x 4 = 160 agen yang mengemulasi 1.600 user (di mana 1 agen mengemulasi 10 user dengan *multi threading*).

Gambar 4.19. Agen yang Digunakan untuk Uji Beban

Skenario yang digunakan dapat dilihat pada Gambar 4.10.

Gambar 4.20. Skenario UjiBebanFRS

Skenario pada Gambar 4.20 dieksekusi menggunakan varian data sebanyak 10 userid/ password yang dapat dilihat pada Gambar 4.21 dan Gambar 4.22.

Gambar 4.21. Varian Skenario UjiBebanFRS Sebanyak 10 Parameter usname

Gambar 4.22. Varian Skenario UjiBebanFRS Sebanyak 10 Parameter pass

Setelah itu diorder ke 160 agen seperti terlihat pada Gambar 4.23 dan 4.24. Pada Gambar 4.23 tersebut tampak order ke setiap agen masing-masing 10 thread (*virtual user* – VU) dengan aksi sebanyak 10x dan waktu jeda (*think time*) 1 *mili second*.

Gambar 4.23. Order Skenario ke Agen Uji

Selain itu juga ada order detail yang menyatakan detail instruksi yang akan dijalankan oleh setiap agen (lihat Gambar 4.24).

Gambar 4.24. Order Detail Skenario ke *Virtual User* pada Agen Uji

Pada saat *schedule* tercapai, maka secara bersamaan seluruh agen uji melakukan pengujian sesuai dengan order masing-masing. Server dimonitor selama pengujian dengan hasil seperti pada Gambar 4.25 dan 4.26.
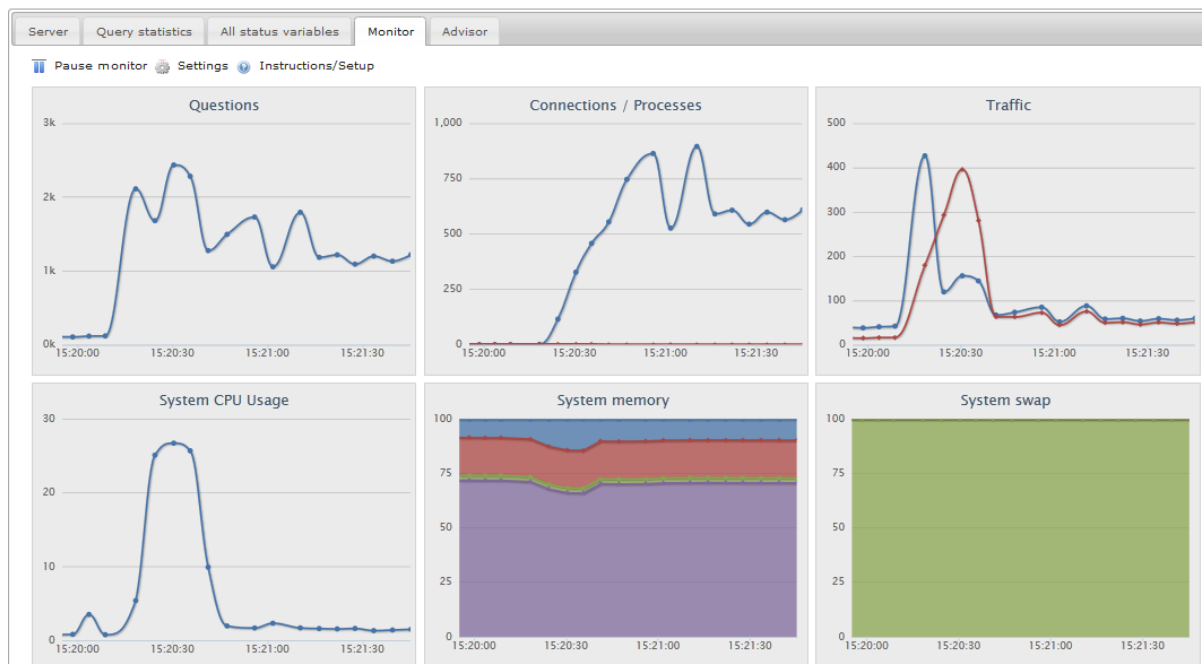
Gambar 4.25. Monitoring Sever Saat Mulai Pengujian Beban

Pada Gambar 4.25 tampak dalam waktu beberapa detik, terjadi trafik yang cukup besar dan koneksi bersama mencapai 800 *concurent connection*.

Gambar 4.26. Monitoring Server Pada Saat Beban Puncak

Pada Gambar 4.26 tercapai puncak koneksi bersama sebanyak 1.142 *concurent connection*. Dengan demikian tampak agen yang mengemulasi 1.600 *concurent connection* telah berhasil mendekati beban puncak di sisi server.

Selain itu juga dilakukan pengujian pembanding mengemulasikan 50 *virtual user* masing-masing sebanyak 10x siklus. Hasilnya dapat dilihat pada Gambar 4.27.
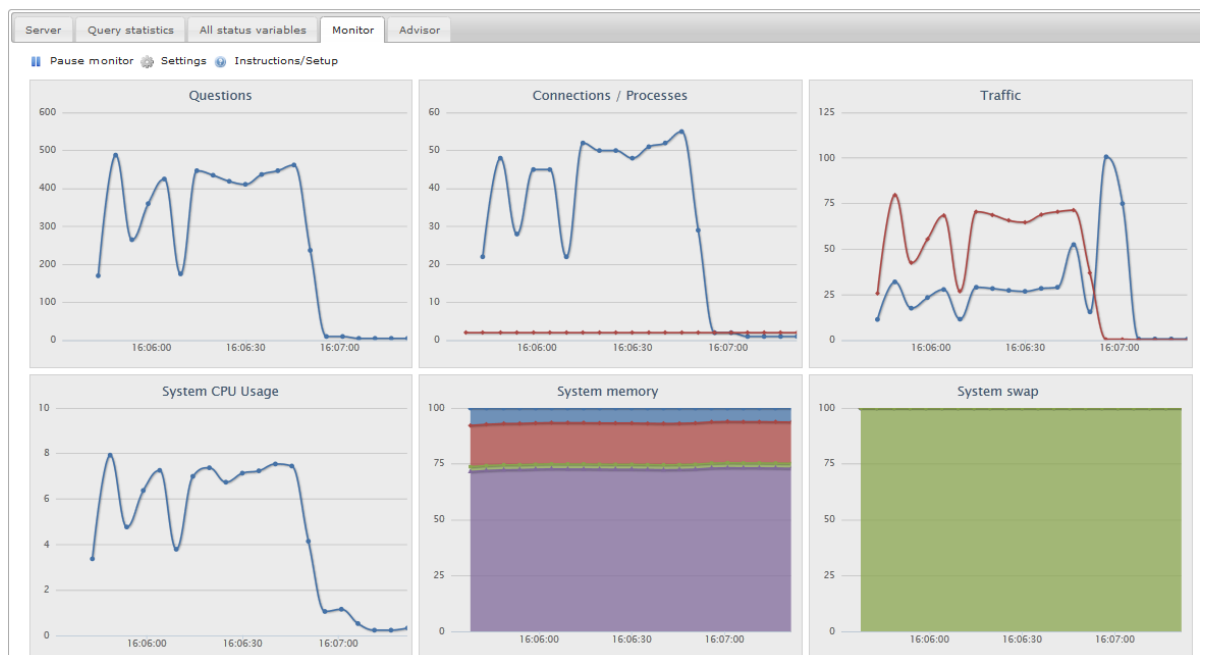
Gambar 4.27. Monitoring Server Pada Percobaan 50 *Virtual User*

Selain monitoring di server, semua hasil pengujian di log ke dalam sistem. Pada Gambar 4.28 tampak log detail dari pengujian suatu agen.

Menu:	Menu:										
		42	AgenEkonomi01	UjiBebanFRS	Finish	2	20	5	20131217091258	20131217091434	10.1000	Detail
About	Scenario	43	AgenEkonomi01	UjiBebanFRS	Finish	10	10	1	20131217095501	20131217095558	84.1620	Detail
	Agent	45	AgentFisip01	UjiBebanFRS	Finish	20	5	2	20131217095502	20131217095555	100.9560	Detail
User		46	ServerLT	UjiBebanFRS	Finish	15	10	4	20131217095504	20131217095823	35.0667	Detail
	Order	50	Lab01081	UjiBebanFRS	Finish	10	10	5	20131217115845	20131217120254	32.5380	Detail
Project		51	Lab01081	UjiBebanFRS	Finish	10	10	2	20131217121348	20131217122644	320.9754	Detail
	Execute	52	Lab01082	UjiBebanFRS	Finish	10	10	1	20131217121347	20131217122602	353.2632	Detail

List of Test State

Menu:	Menu:	No	VU	Frq	StartTime	EndTime	Result	Duration (ms)
About	Scenario							
User	Agent	1	1	1	20131217041702	20131217041704	{null=[HTTP/1.1 200 OK], ETag=["290341-57e-4c4c45a1803c0"], Date=[Tue, 17 Dec 2013 09:05:34 GMT], Content-Length=[1406], Last-Modified=[Sat, 14 Jul 2012 06:31:51 GMT], Content-Type=[text/html; charset=UTF-8], Accept-Ranges=[bytes], Ser	2091
Project	Order	2	1	1	20131217041705	20131217041706	{null=[HTTP/1.1 200 OK], Date=[Tue, 17 Dec 2013 09:05:36 GMT], Content-Length=[1363], Set-Cookie=[password=deleted; expires=Mon, 17-Dec-2012 09:05:35 GMT, username=deleted; expires=Mon, 17-Dec-2012 09:05:35 GMT], Content-Type=[text/html; charset=UTF-8], C	936
	Execute	3	1	1	20131217041707	20131217041708	{null=[HTTP/1.1 200 OK], Date=[Tue, 17 Dec 2013 09:05:38 GMT], Content-Length=[1363], Set-Cookie=[password=sepuluh; expires=Tue, 17-Dec-2013 10:05:38 GMT, username=7208010; expires=Tue, 17-Dec-2013 10:05:38 GMT], Content-Type=[text/html; charset=UTF-8], C	765
	Result	4	1	1	20131217041709	20131217041712	{null=[HTTP/1.1 200 OK], Date=[Tue, 17 Dec 2013 09:05:43 GMT], Content-Length=[0], Content-Type=[text/html; charset=UTF-8], Connection=[close], X-Powered-By=[PHP/5.3.3], Server=[Apache/2.2.3 (CentOS) DAV/2 PHP/5.3.3 mod_python/3.2.8 Python/2.4.3 mod_ssl/2	3401
		5	1	1	20131217041713	20131217041716	{null=[HTTP/1.1 200 OK], ETag=["290341-57e-4c4c45a1803c0"], Date=[Tue, 17 Dec 2013 09:05:47 GMT], Content-Length=[1406], Last-Modified=[Sat, 14 Jul 2012 06:31:51 GMT], Content-Type=[text/html; charset=UTF-8], Connection=[close], Accept-Ranges=[bytes], Ser	3089
							{null=[HTTP/1.1 200 OK], ETag=["290341-57e-4c4c45a1803c0"], Date=[Tue, 17 Dec 2013	

Gambar 4.28. Log Hasil Pengujian Beban (Global dan Detail)

Berdasarkan log pada Gambar 4.28, maka dari sisi agen uji, grafik dapat dikonstruksi seperti pada Gambar 4.29 dengan sampling waktu per 1 detik.
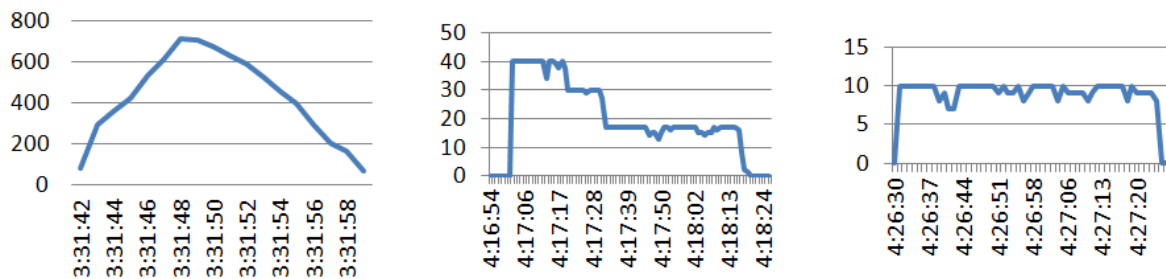
Gambar 4.29. Konstruksi Grafik dari Log Agen

Pada Gambar 4.29 dapat dilihat grafik banyaknya transaksi sukses dari waktu ke waktu pada percobaan 800 virtual user, 40 *virtual user* dan 10 *virtual user*. Grafik ini secara implisit juga menyatakan *concurent connection* yang sukses bertransaksi di server.

Selain performansi dalam bentuk kapasitas, juga diamati performansi dalam bentuk *response time*. Dalam konteks ini diukur waktu yang digunakan untuk mengeksekusi seluruh skenario oleh *virtual user* dalam satu siklus (mulai login sampai logout). Data ini juga dikonstruksi dari log transaksi *virtual user* yang sukses. Untuk kasus 800 *virtual user* hasil percobaannya seperti pada Gambar 4.30.
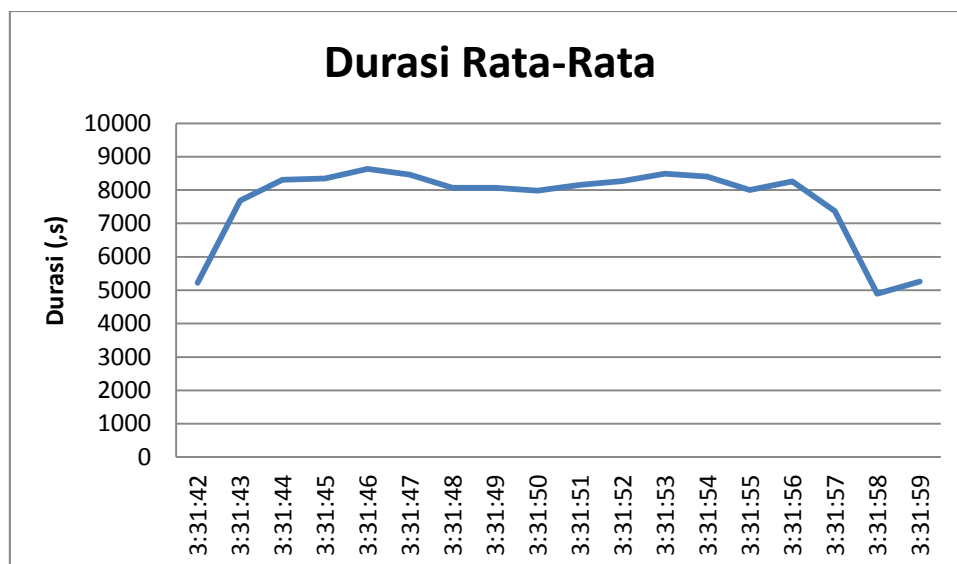
Gambar 4.30. Grafik Durasi Rata-Rata Transaksi Sukses

Pada Gambar 4.30 tampak bahwa durasi transaksi rata-rata antara 5 detik sampai dengan 8 detik untuk mengeksekusi skenario seperti pada Gambar 4.20.

Kesimpulan Hasil Pengujian adalah:

1. Hasil pengujian atas *requirement* no 1, 2 dan 6 diperoleh hasil yang memadai. Uji fungsional menunjukkan aplikasi Student Portal sudah berfungsi dengan baik (no 1 dan 2). Untuk performansi yang diharapkan minimal 500 *concurrent user* telah tercapai dengan pengujian mencapai 1.142 *concurrent user* (maksimal) dan pengujian 800 *concurrent user* yang berjalan baik. Selain itu juga performa yang dimiliki antara 5 sampai dengan 8 detik untuk menjalankan seluruh task secara kontinu. Hal ini merupakan capaian dari sistem Student Portal.

2. Khusus untuk pengujian atas requirement no 5 prosedur pengendalian, berdasarkan *assesment* yang dilakukan, hasilnya bahwa pemenuhan terhadap AC 1 sudah memadai untuk perguruan tinggi minimal di level 3. Sedangkan untuk AC 2, 4 dan 6 sudah hampir memenuhi (diatas 2, dibawah 3). Sedangkan untuk AC 3 nilainya masih belum memadai (dibawah 2). Kecualai AC 1, merupakan temuan, oleh karena itu direkomendasikan untuk dilakukan aksi perbaikan, sehingga minimal mencapai level 3 (sesuai *best practice* untuk perguruan tinggi minimal di level 3).

4.4. Pembahasan Hasil Uji

Berdasarkan audit yang dilakukan terhadap aplikasi Student Portal Unpar, dapat dilihat bahwa model yang diusulkan dan *tools* IWTE dapat memberikan hasil yang memadai, baik terhadap kebutuhan fungsional, pengendalian/ tata kelola dan non fungsional khususnya performansi (kapasitas/ *response time*). IWTE dapat digunakan sebagai alat untuk mendapatkan fakta-fakta yang meyakinkan (termasuk log hasil pengujian) khususnya untuk uji fungsional dan uji beban, sedangkan

untuk uji pengendalian dapat dijadikan alat untuk melakukan assesment untuk mendapatkan tingkat kematangan (*maturity level*) dari proses pengendalian/ tata kelola.

Hasil dari uji fungsional menunjukkan bahwa sistem aplikasi telah berfungsi dengan baik sesuai dengan persyaratan. Demikian juga dengan aspek performansi, dapat memberikan keyakinan berapa besar kapasitas (*concurent user*) yang dapat dilayanai dan berapa *response time*-nya. Khusus untuk uji pengendalian, dapat memberikan gambaran berupa *spider diagram* yang menyatakan profil kematangan proses pengendalian aplikasi berdasarkan 6 *application control* dari Cobit 4.1.

Dengan demikian, dapat disimpulkan bahwa model dan *tools* yang diusulkan untuk mengaudit aplikasi web sekala besar dapat berfungsi dengan baik.

BAB VI. KESIMPULAN DAN SARAN

Berikut adalah kesimpulan dari keseluruhan penelitian dan saran penyempurnaan selanjutnya.

6.1. Kesimpulan

Berdasarkan hasil dari keseluruhan tahapan penelitian dapat disimpulkan bahwa:

1. *Framework* audit ISACA lebih berfokus pada aspek tata kelola (*governance*) atau pengendalian dan fungsional. Aspek non fungsional yang mencerminkan sifat aplikasi web sekala besar tidak dibahas secara eksplisit.
2. Untuk mencakup aspek non fungsional, seperti performansi dan reliabilitas, yang sangat penting pada aplikasi web sekala besar, pada penelitiani ini, diadopsi dari ISO/IEC 25010.
3. Model audit aplikasi web sekala besar diusulkan dengan mengkombinasikan *framework* ISACA dan ISO/IEC 25010, sehingga menyediakan jaminan yang lebih komprehensip tidak hanya aspek fungsional dan pengendalian, tapi juga aspek non fungsional dari aplikasi web sekala besar.
4. Dalam mengimplementasikan model yang diusulkan, diperluan alat uji (*tools*) yang memadai. Atas dasar kebutuhan tersebut, maka telah dikembangkan aplikasi IWTE dengan menambahkan modul *control testing* yang mengakomodasi pengujian pengendalian dari framework ISACA.
5. Hasil pengujian pada kasus Student Portal Unpar (emulator) menunjukkan bahwa model dan *tools* yang diusulkan dapat memberikan hasil audit yang memadai.

6.2. Saran

Berdasarkan pengujian yang dilakukan, ada beberapa saran yang perlu ditindaklanjuti:

1. Bagian yang paling berat dalam penelitian ini adalah bagaimana melakukan uji beban (*load/ stress testing*). Hal ini terjadi karena memerlukan perangkat komputer yang cukup banyak untuk mengemulasi *user* yang banyak dengan prilakunya masing-masing. Untuk mengkoordinasikan komputer tempat agen uji diperlukan *deployment* aplikasi agen yang cukup banyak. Untuk itu perlu dicari teknik *deployment* aplikasi agen yang memadai, sehingga proses penyiapan agen uji menjadi lebih mudah. Selain itu, kehandalan server IWTE khususnya modul yang menangani komunikasi/ koordinasi dengan para agen uji perlu ditingkatkan, sehingga mampu menangani agen uji lebih banyak. Penerapan disain server yang handal dengan model asinkronous perlu diterapkan, tidak hanya menangani komunikasi, tapi juga menangai mekanisme pelaporan hasil uji dan bukti-bukti uji dari agen uji ke server.
2. Analisis statistik saat ini menggunakan aplikasi bantuan berupa EXCEL dan PHP MyAdmin. Ke depan disarankan agar *tools* IWTE juga dilengkapi dengan proses statistik hasil dan pengolahannya sampai ke grafik-grafik yang diperlukan.
3. Saat pengujian IWTE dan aplikasi *auditee* (emulator Student Portal Unpar) di-*install* pada mesin yang sama. Dengan demikian terjadi interferensi performansi. Dalam uji riil disarankan agar mesin IWTE dan mesin aplikasi *auditee* berbeda.

Demikianlah laporan penelitian ini disusun semoga dapat memberikan gambaran yang memadai tentang metode dan hasil-hasilnya.

DAFTAR PUSTAKA

1. ISACA, *IT Audit and Assurance Standards and Guidelines*, http://www.isaca.org/Knowledge-Center/Standards/Documents/ALL-IT-Standards-Guidelines-and-*Tools*.pdf, 2013.
2. Mark Adler, CISA, CISM, CIA, CISSP, dkk., *Cobit 4.1: Framework, Control Objective, Management Guidelines, Maturity Models*, IT Governance Institut, 2007.
3. Bryan C. Kissinger, Ph.D., CISA, CISSP, *Information Technology Compliance: Past, Present and Future*, Information System Control Journal, Vol. 1, 2010.
4. Erik Guldentops, CISA, CISM, *Maturity Measurement—First the Purpose, Then the Method*, Information System Control Journal Vol. 4, 2003.
5. ISACA, *Maturity Assesment Tool User Guide*, ISACA, 2009.
6. Norm Kelson, CISA, CGEIT, CPA, *Generic Application Audit/ Assurance Program*, ISACA, 2009.
7. Tommie W. Singleton, Ph.D., CISA, CGEIT, CITP, CPA, *Auditing Applications Part 1*, Information System Control Journal, Vol. 3, 2012.
8. Tommie W. Singleton, Ph.D., CISA, CGEIT, CITP, CPA, *Auditing Applications Part 2*, Information System Control Journal, Vol. 4, 2012.
9. Zabihollah Rezaee, Rick Elam, Ahmad Sharbatoghlie, *Continuous auditing: the audit of the future*, Managerial Auditing Journal, Vol. 16, Iss: 3, 2001, pp.150 – 158.
10. Abu-Musa, A., *Auditing E-business: New challenges for external auditors,* Journal of American Academy of Business, Cambridge, Vol. 4, Iss: 1, 2004,pp. 28-41. Retrieved from http://search.proquest.com/docview/222876019?accountid=31495
11. Junaid M. Shaikh, *E-commerce impact: emerging technology – electronic auditing*, Managerial Auditing Journal, Vol. 20, Iss: 4, 2005, pp.408 – 421
12. Susan Kennedy, CISA, CIW, *Common Web Application Vulnerabilities*, Information System Control Journal, Vol. 4, 2005.
13. Popa, M., *Detection of the security vulnerabilities in web applications*, Informatica Economica, Vol. 13, No. 1, 2009, pp. 127-136.
    Retrieved from http://search.proquest.com/docview/236781502?accountid=31495
14. Colin Watson, CISA, CITP, CISSP, *COBIT Security Baseline Applied to Business Web Applications*, Information System Control Journal, Vol. 4, 2009.
15. Minesce, A. D., *Load Testing of Web Sites*, IEEE - Internet Computing Journal, Juli 2002.
16. Seng, Jia-Lang; Ko, I-Feng; Lin, Binshan. *A Generic Construct Based Workload Model For Web Search*, Information Processing & Management, Sep 2009, Vol. 45 Issue 5, p529-554, 26p; DOI: 10.1016/j.ipm.2009.04.004.
17. Sampath, Sreedevi; Bryce, Renée C. *Improving The Effectiveness of Test Suite Reduction For User-Session-Based Testing of Web Applications,* Information & Software Technology, Jul2012, Vol. 54 Issue 7, p724-738, 15p; DOI: 10.1016/j.infsof.2012.01.007.
18. Gede Karya, *Perangkat Lunak Uji Performansi Dan Kapasitas Situs Web Terotomasi Multi Agen*, Konferensi Nasional Sistem Informasi (KNSI) 2012, STMIK STIKOM Bali, Denpasar, Februari 2012.
19. Gede Karya, *Automation of Web Enterprise Functional Test Using User Interface-Based Intelligent Agent,* The First International Conference on Computational Science and Information Management (ICoCSIM - 2012), Parapat-Toba, Indonesia, 3-5 December 2012.
20. Roger S. Presman, *Software Engineering A Practioner's Approach*, fitfh editon, McGraw-Hill, 2001.

21. David G. Messerschmitt, *Understanding Networked Applications: A First Course*, Morgan Kaufmann, 2000.
22. Gede Karya, *Perangkat Lunak Uji Performansi Dan Kapasitas Situs Web Terotomasi Multi Agen*, Laporan Penelitian, LPPM-UNPAR, 2011.
23. George Coulouris, Jean Dollimore, Tim Kindberg, *Distributed Systems Concepts and Design*, Addison Wesley, 2001.
24. Bruce Eckel, *Thinking in Java*, Second Edtion, Printice-Hall, 2000.
25. Gede Karya, *Pembangkitan Skenario Dan Data Uji Performansi Dan Kapasitas Situs Web Terotomasi Multi Agen Dengan Metode Back Propagation*, Laporan Penelitian, Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM – Unpar), Desember 2011.
26. Anonym, *Mod Security*, http://www.modsecurity.org/, diakses: 6 Agustus 2011.
27. Anonim, *Extending JMeter*, http://www.jajakarta.org/jmeter/1.7/en/extending/ index.html, diakses terakhir tanggal 13 Juni 2012.
28. Gede Karya, Elisati Hulu, *Otomasi Uji Fungsional Web Enterprise*, Laporan Penelitian, Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM – Unpar), Agustus 2012
29. Anonim, *HTTPUnit Cook Book*, http://httpunit.sourceforge.net/doc/cookbook.html, diakses terakhir tanggal 27 Juli 2012.
30. Gede Karya, Elisati Hulu, *Integrated Web Test Environment*, Laporan Penelitian, Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM – Unpar), Desember 2012
31. Anonim, *Selenium Introduction*, http://seleniumhq.org/docs/01_introducing_selenium.html, diakses tanggal 29 Agustus 2012
32. Anonim, *Watir Home Page*, http://watir.com, diakses tanggal 29 Agustus 2012
33. Robert G. Parker, ITAF – A Professional Practices Framework for IT Assurance, ISACA, April 2008.

LAMPIRAN 1 *AUDIT PROGRAMM ASSURANCE STEP*

Audit/Assurance Program Step	COBIT Cross-reference
1. PLANNING AND SCOPING THE AUDIT	
1.1 **Define audit/assurance objectives.** The audit/assurance objectives are high level and describe the overall audit goals.	
1.1.1 Review the audit/assurance objectives in the introduction to this audit/assurance program.	
1.1.2 Modify the audit/assurance objectives to align with the audit/assurance universe, annual plan and charter.	
1.2 **Define boundaries of review.** The review must have a defined scope. The reviewer must understand the operating environment and prepare a proposed scope, subject to a later risk assessment.	
1.2.1 Obtain an understanding of the services provided by the enterprise, including the scope of services and the effect the services have on the enterprise's activities.	
1.2.2 Establish initial boundaries of the audit/assurance review.	
1.2.2.1 Identify limitations and/or constraints affecting the audit/assurance review.	
1.3 **Define assurance.** The review requires two sources of standards. The corporate standards defined in policy and procedure documentation establish the corporate expectations. At minimum, corporate standards should be implemented. The second source, a good-practice reference, establishes industry standards. Enhancements should be proposed to address gaps between the two.	
1.3.1 Obtain company systems development standards, systems development methodology manual, project management standards, project methodology manual, and application or software manual.	
1.3.2 Determine if COBIT and the appropriate systems development framework will be used as a good-practice reference.	
1.4 **Define the change process.** The initial audit approach is based upon the reviewer's understanding of the operating environment and associated risks. As further research and analysis are performed, changes to the scope and approach will result.	
1.4.1 Identify the senior IT audit/assurance resource responsible for the review.	
1.4.2 Establish the process for suggesting and implementing changes to the audit/assurance program, and to the authorizations required.	
1.5 **Define assignment success.** The success factors need to be identified. Communication among the IT audit/assurance team, other assurance teams and the enterprise is essential.	
1.5.1 Identify the drivers for a successful review (this should exist in the assurance function's standards and procedures).	
1.5.2 Communicate success attributes to the process owner or stakeholder, and obtain agreement.	
1.6 **Define audit/assurance resources required.** The resources required are defined in the introduction to this audit/assurance program.	

Audit/Assurance Program Step	CObIT Cross-reference
1.6.1 Determine the audit/assurance skills necessary for the review.	
1.6.2 Determine the estimated total resources (hours) and time frame (start and end dates) required for the review.	
1.7 **Define deliverables** The deliverable is not limited to the final report. Communication between the audit/assurance teams and the process owner is essential to assignment success.	
1.7.1 Determine the interim deliverables, including initial findings, status reports, draft reports, due dates for responses and the final report.	
1.8 **Communications** The audit/assurance process is clearly communicated to the customer/client.	
1.8.1 Conduct an opening conference to discuss the review objectives with the executive responsible for operating systems and infrastructure.	
2. PLANNING THE APPLICATION AUDIT	
2.1 **Planning team**	
2.1.1 Establish the audit/assurance management team to plan the review.	
2.1.1.1 Assign an experienced IT audit and assurance professional and an operational audit and assurance professional as project managers.	
2.1.1.2 Consider knowledge of the business process area and IT operating environment when making assignments.	
2.1.1.3 Assign lead staff to the planning process.	
2.2 **Understand the application.**	
2.2.1 Obtain an understanding of the business and application process environment.	
2.2.1.1 Obtain an understanding of the application's business environment.	
2.2.1.1.1 Ensure that audit/assurance engagement managers meet with the business and IT executives responsible for the application and business processes.	
2.2.1.1.2 Identify the business process and data owners responsible for the application.	
2.2.1.1.3 Obtain an understanding of the strategic and operational significance of the application.	
2.2.1.1.4 Determine if the application has been developed in- house or purchased, and if the application is maintained in-house or has been contracted/outsourced.	
2.2.1.1.5 Obtain an understanding of the warranties and support in the case of a purchased application.	
2.2.1.1.6 Through discussions and a walkthrough of the general business process, obtain an understanding of the business functions performed by the application and the interfaces with other applications; determine where the controls are located within the application and identify application limitations, where possible.	

Audit/Assurance Program Step	COBIT Cross-reference
2.2.1.1.7 Determine how the business process affects the financial statements of the enterprise (direct interface to the general ledger or mission-critical process that, if not operating correctly, could affect the enterprise's financial performance), or operational significance to the enterprise.	
2.2.1.1.8 Determine the regulatory requirements that impact the business process (external examiners, financial reporting requirements, privacy, data security, etc.).	
2.2.1.1.9 Determine known issues with the business process and application from the perspective of other executives.	
2.2.1.2 Obtain an understanding of the application's functionality.	
2.2.1.2.1 Audit/assurance engagement management and lead staff members meet with the business and IT managers responsible for the application and business processes.	
2.2.1.2.2 Through discussions, perform a walkthrough of the business process and application from source entry through output and reconciliation.	
2.2.1.2.3 Determine how the business process affects the financial statements of the enterprise (direct interface to the general ledger or mission-critical process that, if not operating correctly, could affect the enterprise's financial performance), or operational significance to the enterprise.	
2.2.1.2.4 Determine the regulatory requirements that impact the business process (external examiners, financial reporting requirements, privacy, data security, etc.).	
2.2.1.2.5 Determine known issues with the business process and application from the perspective of other executives.	
2.2.1.3 Understand the application's technical infrastructure.	
2.2.1.3.1 Through discussions with senior management responsible for the development, implementation and operations of the application, obtain an understanding and documentation of the following and how they impact the application: • Technical infrastructure (host, client-server or web-based) • Network (intranet, Internet, or extranet), wireless or wired • Transaction processor (CICS or IMS) • Workstation (desktop/handheld/laptop/special devices) • Operating systems (IBM Mainframe, UNIX/LINUX, Windows, proprietary) • Database management systems (Oracle, DB2, IMS, SQL Server, other) • Insourced or outsourced • Real-time, store and forward, and/or batch • Test and development of operating environments	DS5.3 DS5.4 DS5.8 DS5.10 DS5.11
2.2.1.4 Understand the volatility and level of change affecting the application.	
2.2.1.4.1 Interview business management to determine planned changes, history of problem areas and other known operational issues that would affect the scope of the review.	
2.2.1.4.2 Obtain recent systems requests, incident reports and problem logs.	DS5.6

Audit/Assurance Program Step	CObIT Cross-reference
Identify issues that were not identified in meetings with management.	DS8.2
2.2.1.4.3 Evaluate how volatility and change issues affect the scope of the review, and determine if there are identifiable trends for certain issues.	
2.2.2 Obtain a detailed understanding of the application.	

2.2.2.1 Using the information obtained in management interviews and documentation provided by enterprise and IT, obtain and document a detailed understanding of the application. Consider the:
 	
• Source data
  – Manual input
  – Input interfaces from other applications
• Processing cycle
  – Audit trails
  – Error reporting
  – Internal controls and edits
  – Frequency of application processes
  – Dependency of application on processing cycles and other applications
  – System setup parameters
• Data edits
  – Initial edits
  – Data correction
  – Maintenance of master files
• Output
  – Review and reconciliation
  – Reports generated
  – Output interfaces to other applications
  – Report distribution	

2.2.3 Based on the detailed understanding, identify the transactions in the application and business flow.	
2.3 **Risk assessment**	
2.3.1 Perform a risk assessment of the effect the application has on the business, the IT organization and the potential scope of the review.	PO9.2
2.3.1.1 Consider the importance of business processes and transactions.	
2.3.1.2 Consider financial and regulatory requirements.	
2.3.1.3 Prioritize business processes and transactions for evaluation.	
2.4 **Scope**	
2.4.1 Narrow the scope to business processes and transactions to be evaluated in review.	
2.4.2 Determine the operational audit scope and IT audit scope.	
2.4.3 Determine the audit/assurance resources required to perform the review.	
2.4.4 Determine computer-assisted audit techniques (CAATs) that may be required.	
2.4.5 Identify specific business processes and application transactions to be reviewed.	
2.4.6 Establish the proposed scope.	
2.5 **General controls**	

Audit/Assurance Program Step	COBIT Cross-reference
2.5.1 Evaluate general control reviews to determine the level of reliance that can be placed on the installation controls.	
2.5.1.1 Review the results from the following IT audit/assurance assessments: • Physical security • Identity and access management • Incident/problem management • Change management • Operating system configuration • Information security • Network perimeter management • Database management • IT contingency and business contingency planning	AI6 DS4 DS5 DS8 DS10
2.5.1.2 If open audit/assurance findings remain and are considered material in the context of the application audit, consider what expanded review procedures will be required.	
2.6 **Finalize scope**	
2.6.1 Identify the business processes to be reviewed.	
2.6.1.1 Identify the transactions for the business process.	
2.6.1.2 Identify the control objectives for each business process.	
2.6.1.2.1 Identify the controls that address each control objective.	
2.6.1.2.2 Customize the work program for the controls identified and their control description.	
2.6.2 Assign staff based on skill sets to the various processes.	
2.6.3 Determine IT and operational audit/assurance roles, and establish project management.	
3. SOURCE DATA PREPARATION AND AUTHORIZATION	
3.1 **Source data preparation** Audit/assurance objective: Source documents should be prepared by authorized and qualified personnel following established procedures, and should provide for adequate segregation of duties between the origination and approval of these documents and accountability.	
3.1.1 **Source document design** **Control: Source documents are designed in a way that they increase the accuracy with which data can be recorded, control the workflow and facilitate subsequent reference checking. Where appropriate, completeness controls in the design of the source documents are included.**	AC1
3.1.1.1 Assess whether source documents and/or input screens are designed with predetermined coding, choices, etc., to encourage timely completion and minimize the potential for error.	
3.1.2 **Source data procedures** **Control: Procedures for preparing source data entry are documented, and are effectively and properly communicated to appropriate and qualified personnel.[1]**	AC1

---

[1] These procedures establish and communicate required authorization levels (input, editing, authorizing, accepting and rejecting source documents). The procedures also identify the acceptable source media for each type of transaction.

Audit/Assurance Program Step	CoBiT Cross-reference
3.1.2.1 Determine if the design of the system provides for the identification and management of authorization levels.	
3.1.2.1.1 Verify, through inspection of authorization lists, that authorization levels are properly defined for each group of transactions. Observe that authorization levels are properly applied.	
3.1.2.2 Inspect and observe creation and documentation of data preparation procedures, and inquire whether and confirm that procedures are understood and the correct source media are used.	
3.1.2.3 Inquire whether and confirm that the design of the system provides for the use of preapproved authorization lists and related signatures for use in determining that documents have been appropriately authorized.	
3.1.2.4 Inquire whether and confirm that the design of the system encourages review of the forms for completeness and authorization, and identifies situations where attempts to process incomplete and/or unauthorized documents occur.	
3.1.3 **Data entry authorization** **Control: The function responsible for data entry maintains a list of authorized personnel, including their signatures.**	AC1
3.1.3.1 Where required by procedures, verify that adequate segregation of duties between originator and approver exists.	
3.1.3.2 Inquire whether and confirm that a list of authorized personnel and their signatures is maintained by the appropriate departments. Where possible, use automated evidence collection, including sample data, embedded audit modules or CAATs, to trace transactions to verify that the list of authorized personnel is effectively designed to allow/restrict personnel to enter data.	
3.1.3.3 Determine if a separation of duties (SOD) table exists and review for adequate separation of key duties.	
3.1.3.3.1 Inspect documents, trace transactions through the process and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATs, to trace transactions to verify that authorization access controls are effective.	
3.2 **Form design** Audit/assurance objective: Good input form design should be used to minimize errors and omissions.	
3.2.1 **Transaction identifier** **Control: Unique and sequential identifiers (e.g., index, date and time) are automatically assigned to every transaction.**	AC1
3.2.1.1 Inquire whether and confirm that unique and sequential numbers are assigned to each transaction.	
3.2.2 **Source document design** **Control: Source documents include standard components, contain proper instructions for completion and are approved by management.**	AC1
3.2.2.1 Verify that all source documents include standard components, contain proper documentation (e.g., timeliness, predetermined input codes, default values) and are authorized by management.	

Audit/Assurance Program Step	COBIT Cross-reference
3.3 **Error detection** Audit/assurance objective: Errors and irregularities should be detected so they can be reported and corrected.	
3.3.1 **Document error detection** **Control: Documents that are not properly authorized or are incomplete are returned to the submitting originators for correction and recorded in a log to document their return. Logs are reviewed periodically to verify that corrected documents are returned by originators in a timely fashion, and to enable pattern analysis and root cause review.**	AC1
3.3.1.1 Inquire whether and confirm that, once identified, the system is designed to track and report upon incomplete and/or unauthorized documents that are rejected and returned to the owner for correction.	
3.3.1.2 Inquire and confirm whether logs are reviewed periodically, reasons for returned documents are analyzed and corrective action is initiated.	
3.3.1.3 Determine if the corrective action is monitored for its effectiveness.	
4. **SOURCE DATA COLLECTION AND ENTRY**	
4.1 **Data input preparation input** Audit/assurance objective: Data input should be performed in a timely manner by authorized and qualified staff.	AC2
4.1.1 **Source document criteria** **Control: Criteria for defining and communicating for timeliness, completeness and accuracy of source documents are documented.**	
4.1.1.1 Inquire whether and confirm that criteria for timeliness, completeness and accuracy of source documents are defined and communicated.	
4.1.2 **Source document preparation** **Control: Procedures ensure that data input is performed in accordance with the timeliness, accuracy and completeness criteria.**	AC2
4.1.2.1 Inspect documentation of policies and procedures to ensure that criteria for timeliness, completeness and accuracy are appropriately represented.	
4.2 **Correction and reentry of erroneous data** Audit/assurance objective: Correction and resubmission of data that were erroneously input should be performed without compromising original transaction authorization levels.	
4.2.1 **Out-of-sequence and missing source documents** **Control: Use only prenumbered source documents for critical transactions. If proper sequence is a transaction requirement, identify and correct out-of-sequence source documents. If completeness is an application requirement, identify and account for missing source documents.**	AC2
4.2.1.1 Inquire whether and confirm that policies and processes are established to establish criteria for the identification of classes of critical transactions that require prenumbered source documents or other unique methods of identifying source data.	
4.2.1.2 Inquire and confirm whether critical source documents are prenumbered and out-of-sequence numbers are identified and taken into account.	
4.2.1.3 Identify and review out-of-sequence numbers, gaps and duplicates using automated tools (CAATs).	

Audit/Assurance Program Step	COBIT Cross-reference
4.2.2 **Data editing** **Control: Access rules define and communicate who can input, edit, authorize, accept and reject transactions, and override errors. Accountability is established through access controls and documented supporting evidence to establish accountability in line with role and responsibility definitions.**	AC2
4.2.2.1 For each major group of transactions, inquire whether and confirm that there is documentation of criteria to define authorization for input, editing, acceptance, rejection and override.	
4.2.2.2 Inspect documents, trace transactions through the process and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATs, to trace transactions to verify that authorization controls are effective and that sufficient evidence is reliably recorded and reviewed.	
4.2.2.3 Identify critical transactions. From that population, select a set of critical transactions. Perform the following steps.	
4.2.2.3.1 Compare the actual state of access controls over transaction input, editing, acceptance, etc. with established criteria, policies or procedures.	
4.2.2.3.2 Inspect whether critical source documents are prenumbered or other unique methods of identifying source data are used.	
4.2.2.3.3 Inspect documentation or walk through transactions to identify personnel who can input, edit, authorize, accept and reject transactions, and override errors.	
4.2.2.3.4 Take a sample of transactions within this set for a specific period, and inspect the source documents for those transactions. Verify that all appropriate source documents are available.	
4.2.3 **Error correction** **Control: Procedures are formally established and documented to correct errors, override errors and handle out-of-balance conditions and to follow up on, correct, approve and resubmit source documents and transactions in a timely manner. These procedures should consider things such as error message descriptions, override mechanisms and escalation levels.**	AC2
4.2.3.1 Inquire and confirm whether documented procedures for the correction of errors, out-of-balance conditions and entry of overrides exist.	
4.2.3.2 Determine that the procedures include mechanisms for timely follow-up, correction, approval and resubmission.	
4.2.3.3 Evaluate the adequacy of procedures addressing error message descriptions and resolution, and override mechanisms.	
4.2.4 **Error correction monitoring** **Control: Error messages are generated in a timely manner as close to the point of origin as possible. The transactions are not processed unless errors are corrected or appropriately overridden or bypassed. Errors that cannot be corrected immediately are logged in an automated suspense log, and valid transaction processing continues. Error logs are reviewed and acted upon within a specified and reasonable period of time.**	AC2
4.2.4.1 Inquire whether and confirm that error messages are generated and communicated in a timely manner, transactions are not processed unless errors are corrected or appropriately overridden, errors that cannot be	

Audit/Assurance Program Step	CoBiT Cross-reference
corrected immediately are logged and valid transaction processing continues, and error logs are reviewed and acted upon within a specified and reasonable period of time.	
4.2.5 **Error condition monitoring** **Control: Errors and out-of-balance reports are reviewed by appropriate personnel, followed up on and corrected within a reasonable period of time, and, where necessary, incidents are escalated for attention by a senior-level staff member. Automated monitoring tools may be used to identify, monitor and manage errors.**	AC2
4.2.5.1 Inquire whether and confirm that reports on errors and out-of-balance conditions are reviewed by appropriate personnel; all errors are identified, corrected and checked within a reasonable period of time; and errors are reported until corrected.	
4.2.5.2 Determine if error reports are distributed to someone other than the originating person.	
4.2.5.3 Inspect error and out-of-balance reports, error corrections, and other documents to verify that errors and out-of-balance conditions are effectively reviewed, corrected, checked and reported until corrected.	
4.3 **Source document retention** Audit/assurance objective: Where appropriate for reconstruction, original source documents should be retained for an appropriate amount of time.	
4.3.1 **Source document retention** **Control: Source documents are safe-stored (either by the enterprise or by IT) for a sufficient period of time in line with legal, regulatory or business requirements.**	AC2
4.3.1.1 Inquire whether and confirm that there are policies and procedures in place to determine document retention policies. Factors to consider in assessing the document retention policy include: • Criticality of the transaction • Form of the source data • Method of retention • Location of retention • Time period for retention • Ease of and availability of document retreival • Compliance and regulatory requirements	
4.3.1.2 For a sample of transaction flows, inquire whether and confirm that retention of source documents is defined and applied in relation to established criteria for source document retention.	
5. **ACCURACY, COMPLETENESS AND AUTHENTICITY CHECKS**	
5.1 **Accuracy of transactions** Audit/assurance objective: Entered transactions should be accurate, complete and valid. Input data should be validated and edited; edit failures should be corrected interactively or sent back for correction as close to the point of origination as possible.	
5.1.1 **Transaction edits** **Control: Transaction data are verified as close to the data entry point as possible and interactively during online sessions. Transaction data, whether people-generated, system-generated or interfaced inputs, are subject to a variety of controls to check for accuracy, completeness and validity. Wherever possible, transaction validation continues after the first error is found. Understandable error messages are immediately generated to enable efficient remediation.**	AC3

Audit/Assurance Program Step	CobiT Cross-reference
5.1.1.1 Inquire whether and confirm that validation criteria and parameters on input data are periodically reviewed, confirmed and updated in a timely, appropriate and authorized manner.	
5.1.1.2 Obtain functional description and design information on transaction data entry. Inspect the functionality and design for the presence of timely and complete checks and error messages. If possible, observe transaction data entry.	
5.1.1.3 Select a sample of source data input processes. Inquire whether and confirm that mechanisms are in place to ensure that the source data input processes have been performed in line with established criteria for timeliness, completeness and accuracy.	
5.1.2 **Transaction accuracy completeness and validity** **Control: Controls ensure accuracy, completeness, validity and compliance with regulatory requirements of data input. Controls may include sequence, limit, range, validity, reasonableness, table look-ups, existence, key verification, check digit, completeness (e.g., total monetary amount, total items, total documents, hash totals), duplicate and logical relationship checks, and time edits. Validation criteria and parameters are subject to periodic reviews and confirmation.**	AC3
5.1.2.1 Obtain functional description and design information on data input controls. Inspect the functionality and design for appropriate controls. Examples of controls include the presence of sequence, limit, range, validity, reasonableness, table look-ups, existence, key verification, check digit, completeness (e.g., total monetary amount, total items, total documents, hash totals), duplication, logical relationship checks and time edits, and transaction cutoffs.	
5.1.2.2 Obtain functional description and design information on transaction data validation.	
5.1.2.3 Select a sample of input source data of source documents. Using inspection, CAATs, or other automated evidence collection and assessment tools, validate that input data are a complete and accurate representation of underlying source documents.	
5.2 **Transaction access control** Audit/assurance objective: Access control and role and responsibility mechanisms should be implemented so that only authorized persons whose duties are appropriately segregated from conflicting functions may input, modify and authorize data.	
5.2.1 **Transaction access control** **Control: Access controls are implemented to assign access based on job function.**	AC3
5.2.1.1 Obtain the results from the latest identity management review, and determine if reliance can be placed on the reviews performed previously.	
5.2.1.2 Determine if requirements for segregation of duties for entry, modification and authorization of transaction data as well as for validation rules have been established.	
5.2.1.2.1 Obtain separation of duties tables that define job function and permitted transactions. Determine that no controls or asset protection principles will be violated due to the transaction access assignments.	
5.2.1.2.2 Inquire whether and confirm that processes and procedures are established for the segregation of duties for entry, modification and	

Audit/Assurance Program Step	CoBiT Cross-reference
approval of transaction data as well as for validation rules. Factors to consider in the assessment of segregation of duties policies include criticality of the transaction system and methods for the enforcement of segregation of duties.	
5.2.1.2.3 For important or critical systems, inspect the data input design to ensure that the authorization controls allow only appropriately authorized persons to input or modify data.	
5.3 **Transaction error reporting** Audit/assurance objective: Transactions failing edit and validation routines should be subject to follow-up procedures to ensure that they are ultimately remediated. Any root cause should be identified and procedures should be modified.	
5.3.1 **Suspending and reporting erroneous transactions** **Control: Transactions failing validation are identified and posted to a suspense file in a timely fashion, and valid transactions are not delayed from processing.**	AC3
5.3.1.1 Inquire whether and confirm that policies and procedures exist for the handling of transactions that fail edit and validation checks.	
5.3.1.2 Inspect error corrections, out-of-balance conditions, entry overrides and other documents to verify that the procedures are followed.	
5.3.2 **Suspended transaction follow-up** **Control: Transactions failing edit and validation routines are subject to appropriate follow-up until errors are remediated. Follow-up includes aging transactions to ensure follow-up and conducting root cause analysis to help adjust procedures and automated controls.**	AC3
5.3.2.1 Inspect error and out-of-balance reports, error corrections, and other documents to verify that errors and out-of-balance conditions are effectively reviewed, corrected, checked and reported until corrected.	
5.3.2.2 Inquire whether and confirm that transactions failing edit and validation routines are subject to appropriate follow-up until they are remediated.	
6. **PROCESSING INTEGRITY AND VALIDITY**	
6.1 **Data integrity and validity** Audit/assurance objective: The integrity and validity of data should be maintained throughout the processing cycle and the detection of erroneous transactions should not disrupt processing of valid transactions.	
6.1.1 **Transaction authorization** **Control: Mechanisms are established and implemented to authorize the initiation of transaction processing and to enforce that only appropriate and authorized applications and tools are used.**	AC4
6.1.1.1 Inquire whether and confirm that transaction processing takes place only after appropriate authorization is given.	
6.1.1.2 For a sample application, inquire whether and confirm that segregation of duties is in place. Verify whether segregation of duties is implemented for entry, modification and approval of transaction data as well as for validation rules.	
6.1.1.3 For a sample of critical transactions processes, test whether access controls prevent unauthorized data entry. With searching tools, identify cases where	

Audit/Assurance Program Step	COBIT Cross-reference
unauthorized personnel are able to input or modify data.	
6.1.1.4 For a sample of critical transactions processes, test whether access controls prevent unauthorized data entry. With searching tools, identify cases where unauthorized personnel are able to input or modify data.	
6.1.2 **Processing integrity** **Control: Processing is completely and accurately performed routinely with automated controls, where appropriate. Controls may include checking for sequence and duplication errors, transaction/record counts, referential integrity checks, control and hash totals, range checks, and buffer overflow.**	AC4
6.1.2.1 Inquire whether and confirm that adjustments, overrides and high-value transactions are promptly reviewed in detail for appropriateness by a supervisor who does not perform data entry. Inspect the audit trail, other documents, plans, policies and procedures to verify that adjustments, overrides and high-value transactions are designed effectively to be promptly reviewed in detail.	
6.1.2.2 Inspect the audit trail, other documents, plans, policies and procedures to verify that adjustments, overrides and high-value transactions are designed effectively to be promptly reviewed in detail. Inspect the audit trail, transactions (or batches), reviews and other documents; trace transactions through the process; and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATS, to verify that supervisor reviews are effective to ensure the validity of adjustments, overrides and high-value transactions in a timely manner.	
6.1.2.3 Review the documentation of the tools and applications to verify that they are applicable and suitable for the task. Where appropriate for critical transactions, review the code to confirm that controls in the tools and applications operate as designed. Reprocess a representative sample to verify that automated tools operate as intended.	
6.1.2.4 For highly critical transactions, set up a test system that operates like the live system. Process transactions in the test system to ensure that valid transactions are processed appropriately and in a timely fashion.	
6.1.2.5 Inspect error messages upon data entry or online processing.	
6.1.2.6 Use automated evidence collection, including sample data, embedded audit modules or CAATS, to verify that valid transactions are processed without interruption. Inspect whether and confirm that invalid transactions are reported in a timely manner.	
6.1.3 **Transaction error processing** **Control: Transactions failing validation routines are reported and posted to a suspense file. Where a file contains valid and invalid transactions, the processing of valid transactions is not delayed and all errors are reported in a timely fashion. Information on processing failures is kept to allow for root cause analysis and help adjust procedures and automated controls.**	AC4
6.1.3.1 Inquire whether and confirm that reconciliation of file totals is performed on a routine basis and that out-of-balance conditions are reported.	
6.1.3.1.1 Inspect reconciliations and other documents, and trace transactions through the process to verify that reconciliations effectively determine whether file totals match or the out-of-balance condition is reported to the appropriate personnel.	

Audit/Assurance Program Step	COBIT Cross-reference
6.1.3.2 Inspect the functional description and design information on transaction data entry to verify whether transactions failing edit and validation routines are posted to suspense files.	
6.1.3.2.1 Verify that suspense files are correctly and consistently produced and that users are informed of transactions posted to suspense accounts.	
6.1.3.2.2 Verify that the processing of transactions is not delayed by data entry or transaction authorization errors. Use automated evidence collection, including sample data, base cases (prepared transactions with an expected outcome), embedded audit modules or CAATS to trace transactions to verify that transactions are processed effectively, valid transactions are processed without interruption from invalid transactions and erroneous transactions are reported.	
6.1.3.3 For a sample of transaction systems, verify that suspense accounts and suspense files for transactions failing edit and validation routines contain only recent errors. Confirm that older failing transactions have been appropriately remediated.	
6.1.3.4 For a sample of transactions, verify that data entry is not delayed by invalid transactions.	
6.1.3.5 For highly critical transactions, set up a test system that operates like the live system. Enter different types of errors.	
6.1.3.6 Determine if transactions failing edit and validation routines are posted to suspense files.	
6.1.3.7 Verify that suspense files are correctly and consistently produced.	
6.1.3.8 Verify that the user is informed of transactions posted to suspense accounts.	
6.1.4 **Error monitoring and follow-up** **Control: Transactions failing validation routines are subject to appropriate follow-up until errors are remediated or the transaction is canceled.**	AC4
6.1.4.1 Analyze a representative sample of error transactions on suspense accounts and files, and verify that transactions failing validation routines are checked until remediation.	
6.1.4.2 Verify that suspense accounts and files for transactions failing validation routines contain only recent errors, confirming that older ones have been appropriately remediated.	
6.1.4.3 Verify that error detection and reporting are timely and complete and that they provide sufficient information to correct the transaction.	
6.1.4.4 Ensure that errors are reported appropriately and in a timely fashion.	
6.1.4.5 Take a sample of data input transactions. Use appropriate automated analysis and search tools to identify cases where errors were identified erroneously and cases where errors were not detected.	
6.1.4.6 Ensure that error messages are appropriate for the transaction flow. Examples of appropriate attributes of messages include understandability, immediacy and visibility.	

Audit/Assurance Program Step	CobiT Cross-reference
6.1.5 **Process flow** **Control: The correct sequence of jobs is documented and communicated to IT operations. Job output includes sufficient information regarding subsequent jobs to ensure that data are not inappropriately added, changed or lost during processing.**	AC4
6.1.5.1 Obtain functional description and design information on data input controls.	
6.1.5.1.1 Inspect the functionality and design for the presence of sequence and duplication errors, referential integrity checks, control, and hash totals.	
6.1.5.1.2 With searching tools, identify cases where errors were identified erroneously and cases where errors were not detected.	
6.1.5.2 Determine whether and confirm that jobs sequence is indicated to IT operations.	
6.1.5.2.1 Inquire whether and confirm that jobs provide adequate instructions to the job scheduling system so data are not inappropriately added, changed or lost during processing. Inspect source documents; trace transactions through the process; and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATS to trace transactions to verify that production job scheduling software is used effectively so that jobs run in the correct sequence and provide adequate instructions to the systems.	
6.1.5.2.2 Inspect source documents; trace transactions through the process; and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATS to trace transactions to verify that production job scheduling software is used effectively so that jobs run in the correct sequence and provide adequate instructions to the systems.	
6.1.6 **Unique transaction identifier** **Control: Each transaction has a unique and sequential identifier (e.g., index, date and time).**	AC4
6.1.6.1 Inquire whether and confirm that every transaction is assigned a unique and sequential number or identifier (e.g., index, date, time).	
6.1.6.1.1 Inspect source documents; trace transactions through the process; and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATS to trace transactions to verify that production job scheduling software is used effectively so that jobs run in the correct sequence and provide adequate instructions to the systems.	
6.1.7 **Audit trails** **Control: The audit trail of transactions processed is maintained. Include date and time of input and user identification for each online or batch transaction. For sensitive data, the listing should contain before-and-after images and should be checked by the business owner for accuracy and authorization of changes made.**	AC4
6.1.7.1 Inquire whether and confirm that the audit trail of transactions processed is maintained, including who can disable or delete the audit trails.	
6.1.7.1.1 Inspect the audit trail and other documents to verify that the audit trail is designed effectively. Use automated evidence collection,	

Audit/Assurance Program Step	COBIT Cross-reference
including sample data, embedded audit modules or CAATS, to trace transactions to verify that the audit trail is maintained effectively.	
6.1.7.1.2 Verify that before-and-after images are maintained and periodically reviewed by appropriate personnel.	
6.1.7.2 Inquire whether and confirm that the transaction audit trail is maintained and periodically reviewed for unusual activity.	
6.1.7.2.1 Verify that the review is done by a supervisor who does not perform data entry. Inspect the audit trail, transactions (or batches), reviews and other documents; trace transactions through the process; and, where possible, use automated evidence collection, including sample data, embedded audit modules or CAATS, to verify that periodic review and maintenance of the audit trail effectively detects unusual activity and supervisor reviews are effective to verify the validity of adjustments, overrides and high-value transactions in a timely manner.	
6.1.7.3 Determine that access to sensitive audit trails is restricted to authorized personnel and that access is monitored.	
6.1.8 **Data integrity during system interruptions** **Control: The integrity of data during unexpected interruptions in data processing is maintained.**	AC4
6.1.8.1 Inquire whether and confirm that utilities are used, where possible, to automatically maintain the integrity of data during unexpected interruptions in data processing.	
6.1.8.1.1 Inspect the audit trail and other documents, plans, policies and procedures to verify that system capabilities are effectively designed to automatically maintain data integrity.	
6.1.8.1.2 Review the records of actual interruptions involving data integrity issues, and verify that appropriate tools were used effectively.	
6.1.9 **Monitoring of high-value and adjustment transactions** **Control: Adjustments, overrides and high-value transactions are reviewed promptly in detail for appropriateness by a supervisor who does not perform data entry.**	AC4
6.1.9.1 Inquire whether and confirm that appropriate tools are used and maintenance of thresholds complies with the security requirements.	
6.1.9.2 Inquire whether and confirm that a supervisor periodically reviews system output and thresholds.	
6.1.9.3 Use automated evidence collection, including sample data, embedded audit modules or CAATS, to trace transactions to verify that the tools work as designed.	
6.1.10 **Reconcile file totals** **Control: A parallel control file that records transaction counts or monetary value as data is processed and then compared to master file data once transactions are posted. Reports are generated to identify out-of-balance conditions.**	AC4
6.1.10.1 Inquire whether and confirm that control files are used to record transaction counts and monetary values, and that the values are compared after posting.	
6.1.10.2 Determine if other file total controls are in use.	

Audit/Assurance Program Step	CoBiT Cross-reference
6.1.10.3 Verify that reports are generated identifying out-of-balance conditions and that the reports are reviewed, approved and distributed to the appropriate personnel.	
7. OUTPUT REVIEW, RECONCILIATION AND ERROR HANDLING	
7.1 **Output review, reconciliation and error handling** Audit/assurance objective: Procedures and associated responsibilities to ensure that output is handled in an authorized manner, delivered to the appropriate recipient and protected during transmission should be established and implemented; verification, detection and correction of the accuracy of output should occur; and information provided in the output should be used.	
7.1.1 **Output retention and handling procedures** **Control: Defined procedures for the handling and retaining of output from IT applications are implemented and communicated, follow defined procedures, and consider privacy and security requirements.**	AC5
7.1.1.1 Review output handling and retention procedures for privacy and security.	
7.1.2 **Data retrieval interfaces** **Control: Data retrieval processes utilize access control security to prevent unauthorized access to data.**	AC5
7.1.2.1 Determine if data retrieval tools including data extract generators, open database connectivity interfaces (to Microsoft® Access and Excel) are restricted to data by job function.	
7.1.2.2 Verify that data retrieval security tools are effective by performing appropriate tests of the controls.	
7.1.3 **Sensitive output monitoring** **Control: Physical inventories of all sensitive output, such as negotiable instruments, are routinely performed and compared with inventory records. Procedures with audit trails to account for all exceptions and rejections of sensitive output documents have been created.**	AC5
7.1.3.1 Review the documentation and ensure that procedures specify that periodic inventories should be taken of key sensitive documents and differences should be investigated.	
7.1.3.2 Inquire whether and confirm that physical inventories of sensitive outputs are taken at appropriate intervals.	
7.1.3.3 Verify that physical inventories of sensitive outputs are compared to inventory records and that any differences are acted upon.	
7.1.3.4 Confirm that audit trails are created to account for all exceptions and rejections of sensitive output documents.	
7.1.3.5 Inspect a representative sample of audit trails using automated evidence collection, if possible, to identify exceptions; verify whether they have been detected and action has been taken.	
7.1.3.6 Take a physical inventory sample, and compare it to the associated audit trails to verify that detection operates effectively.	
7.1.4 **Distribution of sensitive output** **Control: If the application produces sensitive output, the recipients who may receive it are defined and the output is clearly labeled so it is recognizable by people and machines, and is distributed accordingly. Where necessary, the**	AC5

Audit/Assurance Program Step	CoBiT Cross-reference
sensitive data are sent it to special access-controlled output devices.	
7.1.4.1 Inquire whether and confirm that sensitive information is defined, agreed upon by the process owner and treated appropriately. This may include labeling sensitive application output and, where required, sending sensitive output to special access-controlled output devices.	
7.1.4.2 For a sample of sensitive data, search output files and confirm that they are properly labeled.	
7.1.4.3 Review the distribution methods of sensitive information and the access control mechanisms of sensitive output devices.	
7.1.4.4 Verify that the mechanisms correctly enforce preestablished access rights.	
7.1.5 **Control total reconciliation** **Control: Control totals in the header and/or trailer records of the output are balanced to the control totals produced by the system at data entry to ensure completeness and accuracy of processing. If out-of-balance control totals exist, they are reported to the appropriate level of management.**	AC5
7.1.5.1 Review design criteria and confirm that they require the use of integrity-based control processes, such as the use of control totals in header and/or trailer records and the balancing of output back to control totals produced by the system.	
7.1.5.2 Inquire whether and confirm that procedures require that out-of-balance conditions and other abnormalities require prompt investigation and reporting.	
7.1.5.3 Inquire whether and confirm that control totals are properly implemented in header and/or trailer records of output to balance back to control totals produced by the system.	
7.1.5.4 Inquire whether and confirm that detected out-of-balance conditions are reported to the appropriate level of management. Inspect out-of-balance reports. Where possible, use automated evidence collection to look for control total errors, and verify that they were acted upon correctly and in a timely manner.	
7.1.6 **Process validation** **Control: Validation of completeness and accuracy of processing is performed before other operations are executed. If electronic output is reused, validation is performed prior to subsequent processing.**	AC5
7.1.6.1 Inquire whether and confirm that procedures have been developed to ensure that output is reviewed for reasonableness, accuracy or other criteria established by the process owner prior to use.	
7.1.6.2 Inquire whether and confirm that procedures have been designed to ensure that the completeness and accuracy of application output are validated prior to the output being used for subsequent processing, including use in end-user processing.	
7.1.6.3 Obtain a list of all electronic outputs that are reused in end-user applications. Verify that the electronic output is tested for completeness and accuracy before the output is reused and reprocessed.	
7.1.6.4 Select a representative sample of electronic output, and trace selected documents through the process to ensure that completeness and accuracy are	

Audit/Assurance Program Step	CoBiT Cross-reference
verified before other operations are performed.	
7.1.6.5 Reperform completeness and accuracy tests to validate that they are effective.	
7.1.7 **Business owner output review** **Control: Procedures to ensure that the business owners review the final output for reasonableness, accuracy and completeness are defined and implemented, and output is handled in line with the applicable confidentiality classification. Potential errors are reported and logged in an automated, centralized logging facility, and errors are addressed in a timely manner.**	AC5
7.1.7.1 Inquire whether and confirm that detected out-of-balance conditions are reported, reports have been designed into the system and procedures have been developed to ensure that reports are provided to the appropriate level of management.	
7.1.7.2 Assess whether procedures have been defined that require the logging of potential errors and their resolution prior to distribution of the reports.	
7.1.7.3 Inquire whether and confirm that output is reviewed for reasonableness and accuracy.	
7.1.7.4 Select a representative sample of output reports, and test the reasonableness and accuracy of the output. Verify that potential errors are reported and centrally logged.	
7.1.7.5 Select a sample of representative transactions, and verify that errors are identified and addressed in a timely manner.	
7.1.7.6 Inspect error logs to verify that errors are effectively addressed in a timely manner.	
8. TRANSACTION AUTHENTICATION AND INTEGRITY	
8.1 **Transaction authentication and integrity** Audit/assurance objective: Before passing transaction data between internal applications and business/operational functions (inside or outside the enterprise), transactions should be checked for proper addressing, authenticity of origin and integrity of content. Authenticity and integrity should be maintained during transmission or transport.	
8.1.1 **Data exchange standards** **Control: Where transactions are exchanged electronically, an agreed-upon standard of communication and mechanisms necessary for mutual authentication is established, including how transactions will be represented, the responsibilities of both parties and how exception conditions will be handled.**	AC6
8.1.1.1 Inquire whether and confirm that a process has been designed to ensure that, for critical transactions, appropriate agreements have been made with counterparties that include communication and transaction presentation standards, responsibilities, authentication, and security requirements.	
8.1.1.2 Select a sample of counterparty agreements for critical transactions and verify that they are complete.	
8.1.1.3 Inquire whether and confirm that systems are designed to incorporate appropriate mechanisms for integrity, authenticity and nonrepudiation, such as adoption of a secure standard or one that is independently verified.	
8.1.1.4 Review documentation and perform a walkthrough to identify applications that are critical for transaction authenticity, integrity and nonrepudiation. For these applications, inquire whether and confirm that an appropriate mechanism for	

Audit/Assurance Program Step	COBIT Cross-reference
integrity, authenticity and nonrepudiation is adopted (i.e., a secure standard or one that is independently verified).	
8.1.1.5 Perform a walkthrough of the code of a sample of applications to confirm that this specification and design are applied. Verify that these specifications have been tested with good results.	
8.1.1.6 Review error logs for transactions that failed authentication, and verify the cause.	
8.1.1.7 Perform a walkthrough of the code of a sample of applications to confirm that specifications for authenticity have been applied. Verify that these specifications have been tested with good results.	
8.1.2 **Tag output** **Control: Tag output from transaction-processing applications in accordance with industry standards to facilitate counterparty authentication, provide evidence of nonrepudiation and allow for content integrity verification upon receipt by the downstream application.**	AC6
8.1.2.1 Obtain and inspect agreements made with counterparties for critical transactions, and ensure that the agreements specify requirements for communication and transaction presentation standards, responsibilities, and authentication and security requirements.	
8.1.2.2 Inquire whether and confirm that systems are designed to incorporate industry standard output tagging to identify authenticated information.	
8.1.2.3 Inspect application manuals and documentation for critical applications to confirm that text regarding specifications and design states that output is appropriately tagged with authentication information.	
8.1.2.4 Select a representative sample of transactions, and verify that authenticity and integrity information is correctly carried forward throughout the processing cycle.	
8.1.2.5 Select a sample of authentication failures to verify that the counterparty agreements operate effectively.	
8.1.3 **Transaction integration with interfacing applications** **Control: Input received from other transaction-processing applications is analyzed to determine authenticity of origin and the maintenance of the integrity of content during transmission.**	AC6
8.1.3.1 Inspect manuals and documentation for critical applications to confirm that design specifications require that input be appropriately verified for authenticity.	
8.1.3.2 Inquire whether and confirm that systems are designed to identify transactions received from other processing applications, and analyze that information to determine authenticity of origin of the information and whether integrity of content was maintained during transmission.	
8.1.3.3 Review error logs for transactions that failed authentication and verify the cause.	

LAMPIRAN 2 *APPLICATION CONTROL MATRIX*

COBIT Control Practice	Assessed Maturity	Target Maturity	Comments
AC1 Source Data Preparation and Authorization 1. Design source documents in a way that they increase accuracy with which data can be recorded, control the workflow and facilitate subsequent reference checking. Where appropriate, include completeness controls in the design of the source documents. 2. Create and document procedures for preparing source data entry, and ensure that they are effectively and properly communicated to appropriate and qualified personnel. These procedures should establish and communicate required authorisation levels (input, editing, authorising, accepting and rejecting source documents). The procedures should also identify the acceptable source media for each type of transaction. 3. Ensure that the function responsible for data entry maintains a list of authorised personnel, including their signatures. 4. Ensure that all source documents include standard components and contain proper documentation (e.g., timeliness, predetermined input codes, default values) and are authorised by management. 5. Automatically assign a unique and sequential identifier (e.g., index, date and time) to every transaction. 6. Return documents that are not properly authorised or are incomplete to the submitting originators for correction, and log the fact that they have been returned. Review logs periodically to verify that corrected documents are returned by originators in a timely fashion, and to enable pattern analysis and root cause review.			

COBIT Control Practice	Assessed Maturity	Target Maturity	Comments
AC2 Source Data Collection and Entry 1. Define and communicate criteria for timeliness, completeness and accuracy of source documents. Establish mechanisms to ensure that data input is performed in accordance with the timeliness, accuracy and completeness criteria. 2. Use only prenumbered source documents for critical transactions. If proper sequence is a transaction requirement, identify and correct out-of-sequence source documents. If completeness is an application requirement, identify and account for missing source documents. 3. Define and communicate who can input, edit, authorise, accept and reject transactions, and override errors. Implement access controls and record supporting evidence to establish accountability in line with role and responsibility definitions. 4. Define procedures to correct errors, override errors and handle out-of-balance conditions, as well as to follow up, correct, approve and resubmit source documents andtransactions in timely manner. These procedures should consider things such as error message descriptions, override mechanisms and escalation levels. 5. Generate error messages in a timely manner as close to the point of origin as possible. The transactions should not be processed unless errors are corrected or appropriately overridden or bypassed. Errors that cannot be corrected immediately should be logged in an automated suspense log, and valid transaction processingshould continue. Error logs should be reviewed and acted upon within a specified and reasonable period of time. 6. Ensure that errors and out-of-balance reports are reviewed by appropriate personnel, followed up and corrected within a reasonable period of time, and that, where necessary, incidents are raised for more senior attention. Automated monitoring tools should be used to identify, monitor and manage errors. 7. Ensure that source documents are safe-stored (either by the business or by IT) for a sufficient period of time in line with legal, regulatory or business requirements.			

COBIT Control Practice	Assessed Maturity	Target Maturity	Comments
AC3 Accuracy, Completeness and Authenticity Checks 1. Ensure that transaction data are verified as close to the data entry point as possible and interactively during online sessions. Ensure that transaction data, whether people-generated, system-generated or interfaced inputs, are subject to a variety of controls to check for accuracy, completeness and validity. Wherever possible, do not stop transaction validation after the first error is found. Provide understandable error messages immediately such that they enable efficient remediation. 2. Implement controls to ensure accuracy, completeness, validity and compliancy to regulatory requirements of data input. Controls may include sequence, limit, range, validity, reasonableness, table look-ups, existence, key verification, check digit, completeness (e.g., total monetary amount, total items, total documents, hash totals), duplicate and logical relationship checks, and time edits. Validation criteria and parameters should be subject to periodic reviews and confirmation. 3. Establish access control and role and responsibility mechanisms so that only authorised persons input, modify and authorise data. 4. Define requirements for segregation of duties for entry, modification and authorisation of transaction data as well as for validation rules. Implement automated controls and role and responsibility requirements. 5. Report transactions failing validation and post them to a suspense file. Report all errors in a timely fashion, and do not delay processing of valid transactions. 6. Ensure that transactions failing edit and validation routines are subject to appropriate follow-up until errors are remediated. Ensure that information on processing failures is maintained to allow for root cause analysis and help adjust procedures and automated controls.			

COBIT Control Practice	Assessed Maturity	Target Maturity	Comments
AC4 Processing Integrity and Validity 1. Establish and implement mechanisms to authorise the initiation of transaction processing and to enforce that only appropriate and authorised applications and tools are used. 2. Routinely verify that processing is completely and accurately performed with automated controls, where appropriate. Controls may include checking for sequence and duplication errors, transaction/record counts, referential integrity checks, control and hash totals, range checks, and buffer overflow. 3. Ensure that transactions failing validation routines are reported and posted to a suspense file. Where a file contains valid and invalid transactions, ensure that the processing of valid transactions is not delayed and that all errors are reported in a timely fashion. Ensure that information on processing failures is kept to allow for root cause analysis and help adjust procedures and automated controls, to ensure early detection or to prevent errors. 4. Ensure that transactions failing validation routines are subject to appropriate follow-up until errors are remediated or the transaction is cancelled. 5. Ensure that the correct sequence of jobs has been documented and communicated to IT operations. Job output should include sufficient information regarding subsequent jobs to ensure that data are not inappropriately added, changed or lost during processing. 6. Verify the unique and sequential identifier to every transaction (e.g., index, date and time). 7. Maintain the audit trail of transactions processed. Include date and time of input and user identification for each online or batch transaction. For sensitive data, the listing should contain before and after images and should be checked by the business owner for accuracy and authorisation of changes made. 8. Maintain the integrity of data during unexpected interruptions in data processing with system and database utilities. Ensure that controls are in place to confirm data integrity after processing failures or after use of system or database utilities to resolve operational problems. Any changes made should be reported and approved by the business owner before they are processed. 9. Ensure that adjustments, overrides and high-value transactions are reviewed promptly in detail for appropriateness by a supervisor who does not perform data entry. 10. Reconcile file totals. For example, a parallel control file that records transaction counts or monetary value as data should be processed and then compared to master file data once transactions are posted. Identify, report and act upon out-of-balance conditions.			

COBIT Control Practice	Assessed Maturity	Target Maturity	Comments
AC5 Output Review, Reconciliation and Error Handling 1. When handling and retaining output from IT applications, follow defined procedures and consider privacy and security requirements. Define, communicate and follow procedures for the distribution of output. 2. At appropriate intervals, take a physical inventory of all sensitive output, such as negotiable instruments, and compare it with inventory records. Create procedures with audit trails to account for all exceptions and rejections of sensitive output documents. 3. Match control totals in the header and/or trailer records of the output to balance with the control totals produced by the system at data entry to ensure completeness and accuracy of processing. If out-of-balance control totals exist, report them to the appropriate level of management. 4. Validate completeness and accuracy of processing before other operations are performed. If electronic output is reused, ensure that validation has occurred prior to subsequent uses. 5. Define and implement procedures to ensure that the business owners review the final output for reasonableness, accuracy and completeness, and that output is handled in line with the applicable confidentiality classification. Report potential errors, log them in an automated, centralised logging facility, and address errors in a timely manner. 6. If the application produces sensitive output, define who can receive it, label the output so it is recognisable by people and machines, and implement distribution accordingly. Where necessary, send it to special access-controlled output devices.			
AC6 Transaction Authentication and Integrity 1. Where transactions are exchanged electronically, establish an agreed-upon standard of communication and mechanisms necessary for mutual authentication, including how transactions will be represented, the responsibilities of both parties and how exception conditions will be handled. 2. Tag output from transaction processing applications in accordance with industry standards to facilitate counterparty authentication, provide evidence of non-repudiation, and allow for content integrity verification upon receipt by the downstream application. 3. Analyse input received from other transaction processing applications to determine authenticity of origin and the maintenance of the integrity of content during transmission.			

LAMPIRAN 3 *CONTROL ASSESMENT FORM TEMPLATE*

Berikut adalah template form assesment pengendalian berdasarkan Cobit 4.1. Pada template ini hanya ditampilkan sampai level 3.

Process	PO1 Define a Strategic IT Plan

Management of the process of *Define a strategic IT plan* that satisfies the business requirement for IT of sustaining or extending the business strategy and governance requirements while being transparent about benefits, costs and risks is:

Assessment Status	Open

LINK | Back to Assessment Overview |

PO1	Define a Strategic IT Plan		
Level	Compliance	Contribution	Value
0	0,00	0,00	0,00
1	0,00	1,00	0,00
2	0,00	1,00	0,00
3	0,00	1,00	0,00
4	0,00	1,00	0,00
5	0,00	1,00	0,00

	Maturity Level =	0,00

Instructions: A relative Weight between 0 and 10 should be allocated for each statement, and then an 'x' is used to indicate which statement is applicable.

Maturity Level	0 Non-existent

Do you agree... (Not at all / A little / To some degree / Completely / Relative Importance)

Nr	Statement	Weight		Relative Importance
1	IT strategic planning is not performed.	5		0,00
2	There is no management awareness that IT strategic planning is needed to support business goals.	5		0,00
	Total Weight	10		

Maturity Level	1 Initial/*Ad Hoc*

Do you agree...

Nr	Statement	Weight		
1	The need for IT strategic planning is known by IT management.	5		0,00
2	IT planning is performed on an as-needed basis in response to a specific business requirement.	5		0,00
3	IT strategic planning is occasionally discussed at IT management meetings.	5		0,00
4	The alignment of business requirements, applications and technology takes place reactively rather than by an organisationwide strategy.	5		0,00
5	The strategic risk position is identified informally on a project-by-project basis.	5		0,00
	Total Weight	25		

Maturity Level	2 Repeatable but Intuitive

Do you agree...

Nr	Statement	Weight		
1	IT strategic planning is shared with business management on an as-needed basis.	5		0,00
2	Updating of the IT plans occurs in response to requests by management.	5		0,00
3	Strategic decisions are driven on a project-by-project basis without consistency with an overall organisation strategy.	5		0,00
4	The risks and user benefits of major strategic decisions are recognised in an intuitive way.	5		0,00
	Total Weight	20		

Maturity Level	3 Defined

Do you agree...

Nr	Statement	Weight		
1	A policy defines when and how to perform IT strategic planning.	5		0,00
2	IT strategic planning follows a structured approach that is documented and known to all staff.	5		0,00
3	The IT planning process is reasonably sound and ensures that appropriate planning is likely to be performed.	5		0,00
4	However, discretion is given to individual managers with respect to implementation of the process, and there are no procedures to examine the process.	5		0,00
5	The overall IT strategy includes a consistent definition of risks that the organisation is willing to take as an innovator or follower.	5		0,00
6	The IT financial, technical and human resources strategies increasingly influence the acquisition of new products and technologies.	5		0,00
7	IT strategic planning is discussed at business management meetings.	5		0,00
	Total Weight	35		

LAMPIRAN 4 APLIKASI EMULATOR STUDENT PORTAL

Aplikasi Portal Akademik merupakan emulasi dari StudentPortal Unpar, khusus berkaitan dengan pendaftaran rencana studi di setiap awal semester, yang sering disebut kegiatan pendaftaran Form Rencana Studi (FRS). Aplikasi ini dapat diakses melalui situs http://centeris.unpar.ac.id/sample/.
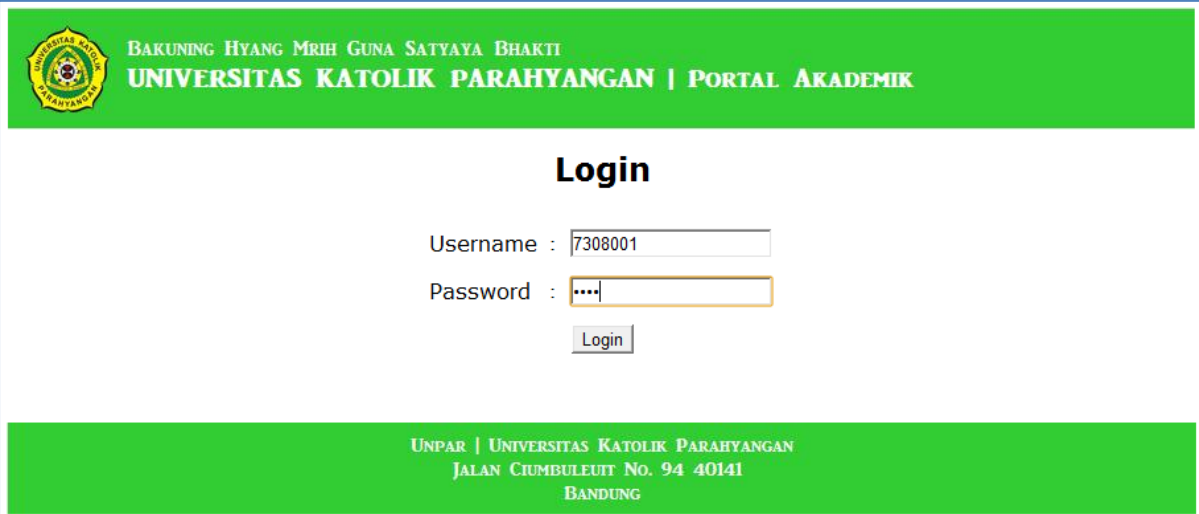
Aplikasi ini memiliki fitur:

1. Halaman depan, yang menampilkan judul dan link ke fitur login. Untuk lebih jelas dapat dilihat pada gambar 4.1.
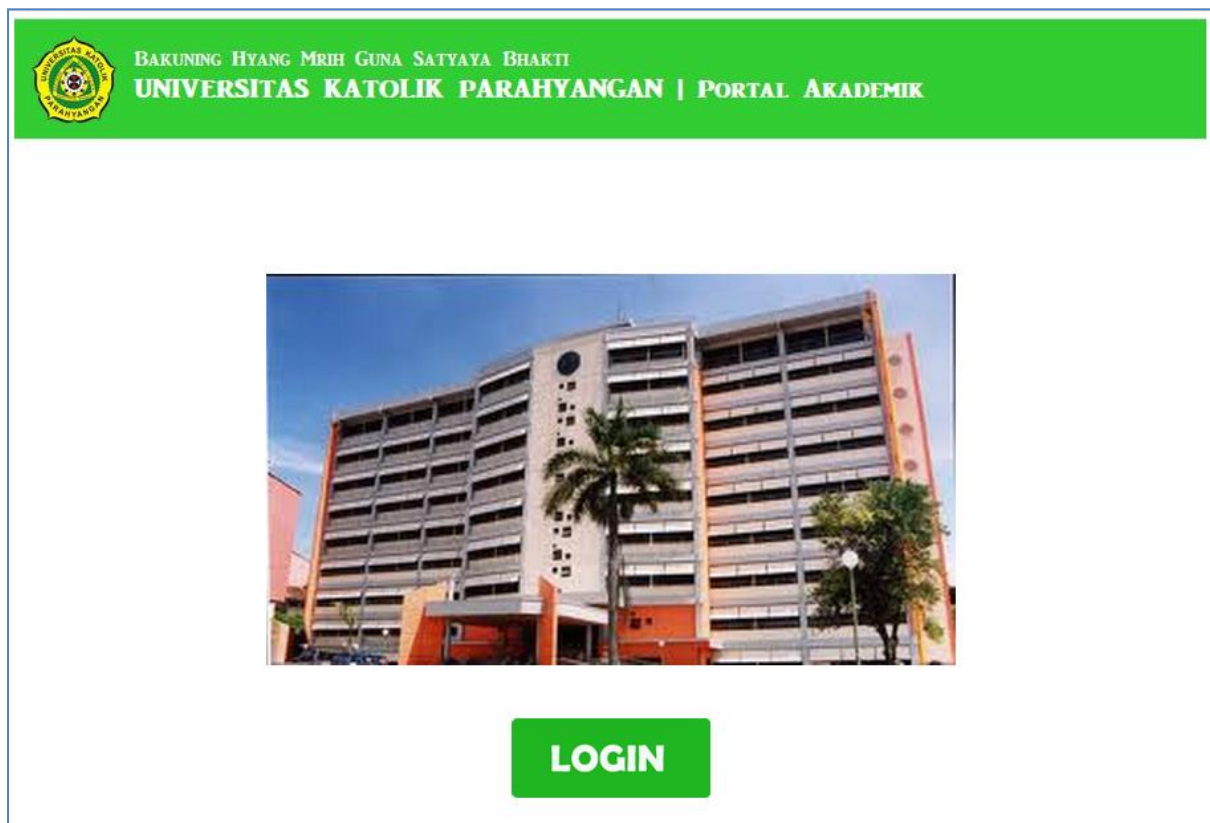
Gambar 4.1. Halaman Depan Aplikasi Portal Akademik

Pada gambar 4.1. dapat dilihat ada tombol "LOGIN". Jika diklik akan menampilkan halaman Login.

2. Login, menampilkan form untuk login, berupa isian userid dan password (lihat gambar 4.2).

Gambar 4.2. Halaman Login Aplikasi Portal Akademik

Pada gambar 4.2, jika userid dan password yang dimasukkan terdaftar (contoh: userid=" 7308001" dan password = "satu"), setelah diklik tombol login maka akan menampilkan halaman awal registrasi FRS (gambar 4.4). Akan tetapi jika userid dan password yang dimasukkan tidak terdaftar (tidak valid) maka akan menampilkan pesan seperti pada gambar 4.3.
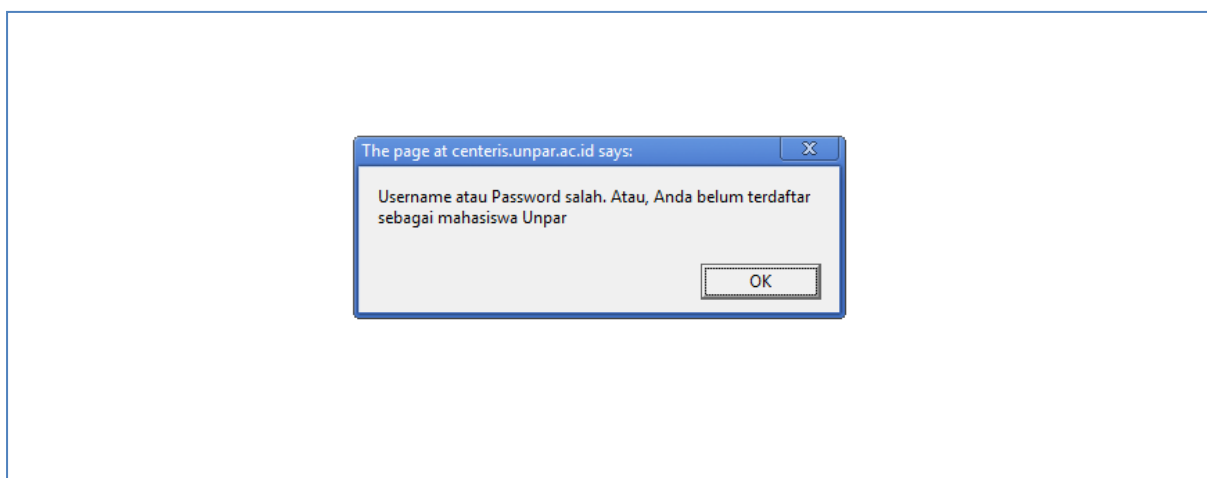
Gambar 4.3. Pesan Jika Login Tidak Berhasil

Pada gambar 4.3, jika tombol OK diklik, maka akan kembali ke halaman Login (gambar 4.2).

3. Halaman Registrasi FRS. Fitur ini menyediakan fungsi untuk memasukkan mata kuliah yang akan diambil pada saat FRS. Terdiri atas halaman awal, seperti pada gambar 4.4.

Gambar 4.4. Halaman Awal Registrasi FRS

Pada gambar 4.4. terlihat ada informasi semester yang sedang berlangsung, yaitu: Ganjil 2011/2012. Selain itu juga ditampilkan identitas dari mahasiswa yang telah Login yaitu: Nurisya Hafisah dengan NPM 2008730001 yang saat ini mejadi mahasiswa di program studi Teknik Informatika. Selain itu juga ada link ke informasi lain seperti tata tertib ujian, data akademik yang mencakup: Rencana Studi, Jadwal baik kuliah, UTS dan UAS, Nilai dan Indeks Prestasi, serta Tagihan yang berisi informasi keuangan, serta tombol logout untuk keluar dari aplikasi.

Untuk melaksanakan pendaftaran, maka mahasiswa harus mengklik link "Rencana Studi". Jika link tersebut dipilih akan ditampilkan halaman registrasi seperti pada gambar 4.5.

Pada gambar 4.5. terlihat bahwa ada bagian isian Rencana Studi. Pada bagian atas ditampilkan indeks prestasi komulatif (IPK) dari mahasiswa, dalam hal ini 2.78. Atas dasar IPK tersebut sistem juga memberikan informasi batas maksimal pengambilan SKS pada semester ini, yaitu: 21 SKS.

Gambar 4.5. Halaman Isian FRS

Pada bagian di bawahnya ditampilkan semeter saat ini, yaitu: Semester Ganjil, dan ditampilkan daftar mata kuliah yang dapat diambil. Untuk setiap mata kuliah memiliki informasi semester, kode mata kuliah (Kode MK), nama mata kuliah (Nama MK), jumlah SKS dan isian untuk dipilih (Pilih).

Pemilihan mata kuliah dapat dilakukan dengan mengklik check box di bagian Pilih. Untuk setiap pemilihan akan otomatis mengupdate informasi "Jumlah SKS yang diambil adalah ... SKS" pada bagian bawahnya. Untuk lebih jelas dapat dilihat pada gambar 4.6.

Semester: Ganjil

Semester	Kode MK	Nama MK	sks	Pilih
s1	AKS101	Pengantar TIK	3	☑
s1	AKS121	Struktur Diskret	3	☑
s1	AKS141	Dasar-Dasar Pemrograman	3	☑
s1	AMT107	Kalkulus	4	☑
s2	AKB111	Pengantar Basis Data	2	☐
s2	AKS122	Logika Formal Dasar	2	☐
s2	AKS142	Pemrograman Berorietasi Objek	3	☐
s2	AKT121	Sistem Dijital	3	☐
s2	AKT141	Pemrograman Berbasis Web	3	☐
s2	AMT211	Aljabar Linear dan Matriks	3	☐

Jumlah sks yang diambil adalah **13** sks

Lanjutkan

Gambar 4.6. Pemilihan Mata Kuliah dan Update Jumlah SKS Otomatis

Pada gambar 4.6 dapat dilihat bahwa ada 4 mata kuliah yang telah dipilih, yang masing-masing memiliki SKS 3, 3, 3 dan 4. Dengan demikian total SKS yang telah diambil adalah 13 SKS. Informasi ini langsung terupdate pada bagian bawahnya.

Jika semua mata kuliah sudah dipilih, maka dapat dilanjutkan dengan mengklik tombol "Lanjutkan". Setelah itu akan tampil form konfirmasi seperti pada gambar 4.7.

Konfirmasi Mata Kuliah Yang Diambil

Nama : Nurisya Hafisah

NPM : 2008730001

Notice: Undefined variable: total in /home/iqbal/www/sample/konfirmasi.php on line **93**

Kode MK	Nama MK	sks
AKS101	Pengantar TIK	3
AKS121	Struktur Diskret	3
AKS141	Dasar-Dasar Pemrograman	3
AMT107	Kalkulus	4
	Jumlah sks	**13 sks**

Reset ___ Save

Gambar 4.7. Konfirmasi Pengambilan Mata Kuliah

Jika mata kuliah yang dikonfirmasi sudah sesuai dengan yang dipilih, maka registrasi dapat diakhiri dengan mengklik tombol "Save". Jika berhasil, maka akan muncul halaman konfirmasi pendaftaran rencana studi berhasil seperti pada gambar 4.8.
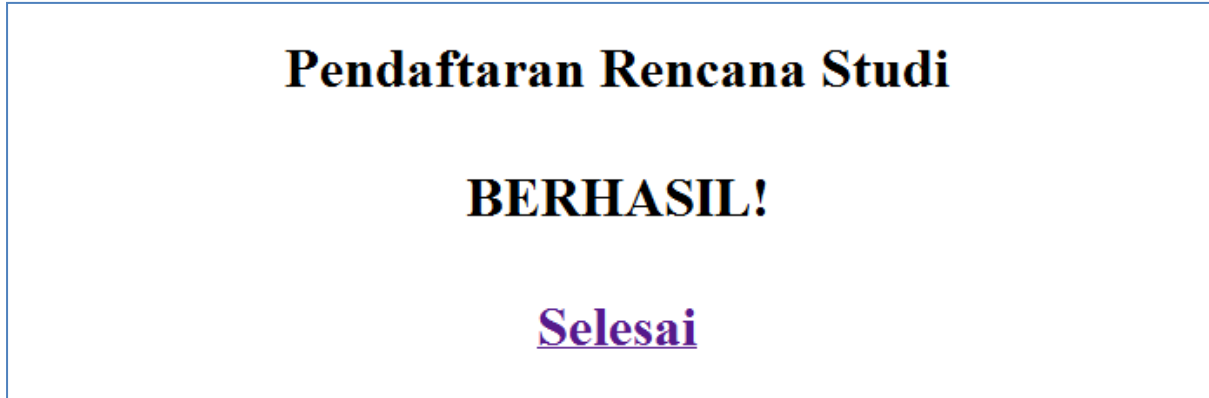
Gambar 4.8. Tanda Pendaftaran Berhasil

Pada gambar 4.8 setelah melihat hasil konfirmasi berhasil, maka proses pendaftaran rencana studi dapat diakhir dengan mengklik link "Selesai". Setelah itu akan ditampilkan kembali halaman awal FRS seperti pada gambar 4.9.

Gambar 4.9. Jika sudah selesai dapat menggunakan Tombol Logout untuk keluar

Pada gambar 4.9, jika ingin keluar, dapat mengklik tombol "Logout". Jika demikian maka akan ditampilkan kembali halaman awal seperti pada gambar 4.1. Dengan demikian proses pendaftaran rencana studi (FRS) selesai dilaksanakan.