LAPORAN KAJIAN

MANAJEMEN PENGAMANAN e-LAYANAN

KEMENTERIAN PENDIDIKAN NASIONAL

TAHUN 2010

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kementerian Pendidikan Nasional (Kemdiknas) adalah satu dari 12 instansi pemerintah yang diprogramkan untuk merintis dan melaksanakan reformasi birokrasi pada tahun 2010-2011. Program yang dikoordinasikan oleh Kantor Menteri Negara Pendayagunaan Aparatur Negara dan Reformasi Birokrasi tersebut merupakan komitmen pemerintah RI dalam melaksanakan Undang-Undang Nomor 28 Tahun 1999 tentang Penyelenggaraan Negara yang Bersih dan Bebas dari Korupsi, Kolusi, dan Nepotisme.

Sesuai dengan Peraturan Menteri Negara Pendayagunaan Aparatur Negara Nomor PER/15/M.PAN/7/2008 tentang Pedoman Umum Reformasi Birokrasi, reformasi birokrasi dilakukan terhadap 3 aspek utama, yaitu kelembagaan, ketatalaksanaan, dan sumber daya manusia. Dalam bidang ketatalaksanaan, khususnya dalam menjamin standar kualitas proses dan data dari layanan secara elektronik, sistem-sistem e-Layanan di lingkungan Kemdiknas perlu distandarkan keamanannya, mulai dari pengamanan terhadap penugasan siapa yang bertanggung jawab terhadap tiap proses pada layanan tersebut sampai dengan pengamanan terhadap catatan dari hasil tiap proses.

Layanan Prima Pendidikan Nasional telah dicanangkan sebagai salah satu agenda Reformasi Birokrasi Internal Kemdiknas, salah satu wujud nyatanya berbentuk layanan elektronik atau disebut juga e-Layanan. Peningkatan layanan pendidikan melalui e-Layanan meski sudah mampu memberikan manfaat, masih mempunyai beberapa kekurangan yang harus dipenuhi untuk mencapai Layanan Prima Pendidikan. Beberapa permasalahan yang dihadapi dalam memberikan Layanan Prima Pendidikan Nasional diantaranya:

- Sebagian besar e-Layanan yang pernah ada pada Kementerian Pendidikan Nasional belum mewadahi layanan dengan transaksi elektronik. E-Layanan yang ada pada Kementerian belum menerapkan transaksi elektronik. Hal ini ditunjukkan dengan proses pengisian form dilakukan disistem, namun untuk proses bisnis tidak ada penerapan transaksi elektronik.
- Banyaknya e-Layanan yang masih besar porsi proses manual-nya.

- Masing-masing e-Layanan mempunyai pengelolaan pengamanan sendiri-sendiri. Setiap sistem informasi yang ada saat ini pada Kementerian Pendidikan Nasional memiliki masing-masing pengelolaan pengamanan. Padahal jika dilihat dari segi fungsi dan proses bisnisnya, pengamanan seharusnya dapat digeneralisasi dan dapat diakses dari berbagai e-Layanan. Sehingga tidak perlu pengguna memiliki banyak login untuk memanfaatkan e-Layanan Kementerian Pendidikan Nasional.
- Transaksi elektronik yang seharusnya dapat memberikan tingkat kepercayaan yang tinggi, dan sebagai akibat kurangnya kemudahan yang diperoleh pengguna maka tingkat kepercayaan terhadap e-Layanan akan berkurang.

Semua permasalahan tersebut diharapkan dapat diselesaikan dengan penerapan manajemen pengamanan pada sistem-sistem e-Layanan di lingkungan Kemdiknas. Dengan banyaknya dan berkembangnya e-Layanan di masa yang akan datang, perlu disusun suatu konsep pengamanan standar yang dapat diterapkan di berbagai e-Layanan di lingkungan Kemdiknas. Agar standar pengamanan aplikasi-aplikasi e-Layanan tersebut mudah dan cepat implementasinya, "Manajemen Pengamanan e-Layanan" tersebut perlu diwujudkan ke dalam suatu komponen perangkat lunak.

1.2 Tujuan

Tujuan pengamanan e-Layanan dapat dibagi menjadi 2 bagian jika dilihat dari periode tujuannya, yaitu jangka pendek dan jangka panjang. Tujuan yang akan dicapai dalam jangka pendek adalah sebagai berikut:

- Adanya jaminan alur pelayanan yang aman.
  Dengan adanya jaminan tersebut, maka yang meningkatkan rasa percaya yang dilayani terhadap yang melayani.
- Adanya jaminan penugasan penanggung jawab kepada pihak yang berwenang.
  Hak akses hanya diberikan kepada orang yang berwenang dalam rangka memberikan layanan yang optimal. Dalam transaksi elektronik, kesalahan pemberian kewenangan oleh sistem harus diminimalisir dengan cara membatasi masa aktif kewenangan tersebut.
- Terciptanya pengamanan terhadap catatan hasil tiap proses.
  Catatan atau log pada sistem sangat penting dalam IT forensik.

Dengan memberikan jaminan keamanan tersebut, tentunya akan sangat membantu dalam mengamankan sistem.

- Menyederhanakan urusan pengguna sistem dengan aplikasi-aplikasi e-Layanan Kemdiknas.

Tujuan yang akan dicapai dalam jangka panjang adalah tercapainya kepuasan pelanggan terhadap e-Layanan.

1.3 Permasalahan

Terkait dengan pemenuhan prinsip-prinsip reformasi birokrasi Kemdiknas, beberapa permasalahan dan kebutuhan sistem telah teridentifikasi untuk didapatkan solusinya dalam kajian ini. Adapun pemetaan prinsip ke dalam kebutuhan sistem dapat dilihat pada Tabel 1.

Tabel 1. Permasalahan dan kebutuhan manajemen pengamanan e-Layanan

No.	Prinsip Reformasi Birokrasi Kemdiknas	Permasalahan/Kebutuhan Sistem
1	Fokus pada kepentingan target layanan melalui interaksi secara self-managed/self-service.	• Jumlah pengguna yang sangat banyak • Ancaman terhadap otentikasi.
2	Berorientasi pada fungsi, tidak terpengaruh perubahan organisasi.	• Independensi otorisasi terhadap struktur organisasi • Independensi tanggungjawab terhadap orang.
3	Berbasis informasi, bukan dokumen fisik.	• Jaminan keabsahan transaksi elektronik, pemenuhan aspek *non-repudiation* • Pengamanan akses terhadap pihak-pihak yang tidak berhak.
4	Berazas prinsip aliran informasi dan pemakaian data bersama, single source of data.	• Pengamanan ruang lingkup data pada level record. Pengguna dapat dibatasi hak aksesnya hanya sebatas data organisasinya saja.

1.4 Metodologi

Lingkup pekerjaan yang ditangani dalam kegiatan ini adalah melakukan kajian manajemen pengamanan e-Layanan di lingkungan Kementerian Pendidikan Nasional. Hasil dari pelaksanaan kegiatan adalah tersusunnya naskah kajian manajemen pengamanan e-Layanan di lingkungan Kementerian Pendidikan Nasional. Untuk menghasilkan output yang diharapkan, pendekatan metodologi yang digunakan terdiri dari beberapa langkah atau fase, yaitu:

a. Identifikasi obyek-obyek dalam e-Layanan yang diamankan
b. Identifikasi jenis-jenis hak dan wewenang bagi para pemangku kepentingan dalam berhubungan dengan e-Layanan
c. Identifikasi bentuk-bentuk ancaman dan penyalahgunaan hak akses
d. Perancangan model konseptual pengamanan aplikasi e-Layanan.

1.5 Sistematika Pembahasan

Laporan kajian manajemen pengamanan e-Layanan ini disusun dengan susunan sebagai berikut:

Bab I Pendahuluan
Bab ini berisi latar belakang, tujuan, sasaran, permasalahan dan metodologi dalam kegiatan manajemen pengamanan e-Layanan.

Bab II Teori Penunjang
Bab ini berisi hasil kajian dari literatur-literatur yang berhubungan dengan manajemen pengamanan e-Layanan. Bab ini berisi teori-teori penunjang yang digunakan dalam penyelesaian kegiatan manajemen pengamanan e-Layanan.

BAB III Analisis
Bab ini membahas kebutuhan akan pengamanan beserta analisis dari permasalahan pengamanan yang dihadapi, ancaman-ancaman dalam pengamanan, serta peluang-peluang untuk melakukan pengamanan.

BAB IV Model Konseptual Pengamanan Aplikasi e-Layanan
Bab IV berisi hasil perancangan awal yang dapat menjadi solusi untuk pengamanan e-Layanan, khususnya dari sisi aplikasinya; apa-apa yang harus dilakukan untuk mengamankan aplikasi dan data dalam e-Layanan, serta hal-hal yang dapat mengurangi permasalahan keamanan yang dihadapi pengguna.

BAB V Kesimpulan

Bab V berisi kesimpulan yang ditarik dari hasil-hasil kajian manajemen pengamanan e-Layanan yang sudah dilakukan.

BAB II
TEORI PENUNJANG

2.1 Definisi Pengamanan dalam e-Layanan

Sistem yang baik adalah sistem yang terjaga dari segala bentuk ancaman yang mengakibatkan sistem tersebut menjadi rusak atau bisa disebut sebagai sistem yang aman. Jadi, pengamanan sebuah sistem adalah segala betuk mekanisme yang harus dijalankan dalam sebuah sistem yang ditujukan akan sistem tersebut terhindar dari segala ancaman yang membahayakan, baik keamanan yang melingkupi data, informasinya ataupun pelaku sistem (user). Keamanan sebuah sistem tidak terjadi begitu saja, tetapi harus dipersiapkan sejak proses pendesignan sistem tersebut.

2.2 Level *Security*

Macam-macam level dalam melakukan security terhadap data adalah sebagai berikut:

- ***Database system level***: merupakan mekanisme autentikasi dan otorisasi untuk mengijinkan pemakai tertentu melakukan akses data yang diperlukan saja.

- ***Operating system level***: Operating system super-user dapat melakukan apapun terhadap database. Kemanan sistem operasi yang handal dan bagus diperlukan dalam hal ini.

- ***Network level***: pada level ini proses kemanan harus menggunakan enkripsi untuk menjaga Eavesdropping (pembacaan yang tidak terotorisasi terhadap pesan-pesan tertentu dan Masquerading (berpura-pura menjadi pemakai yang sah atau mengirimkan pesan yang seolah berasal dari pemakai yang sah).

- ***Physical Level***: melakukan akses fisik terhadap komputer memungkinkan terjadinya perusakan data, kemanan dengan menggunakan kunci yang diperlukan. Komputer juga harus diamankan dari kebanjiran, kebakaran dan lainnya.

- ***Human Level***: Pemakai harus disaring dahulu untuk memastikan bahwa pemakai yang sah tidak memperbolehkan memberikan hak akses kepada orang lain (penyusup). Pemakai harus dilatih dalam pemilihan password dan menjaga kerahasiaannya.

2.3 Celah-celah Keamanan

Dewasa ini permasalahan keamanan tersebut seringkali terjadi. Permasalahan keamanan yang sering terjadi pada aplikasi web adalah sebagai berikut:

2.3.1. Denial of Service

Denial of Service (DoS) attack adalah sebuah usaha (dalam bentuk serangan) untuk melumpuhkan sistem yang dijadikan target sehingga sistem tersebut tidak dapat menyediakan servis-servisnya (denial of service).

Cara untuk melumpuhkan dapat bermacam-macam dan akibatnya juga dapat beragam. Sistem yang diserang dapat menjadi "hang, crash", tidak berfungsi, atau turun kinerjanya (beban CPU tinggi). Serangan denial of service berbeda dengan kejahatan pencurian data atau kejahatan memonitor informasi yang lalu lalang. Dalam serangan DoS tidak ada yang dicuri. Akan tetapi, serangan DoS dapat mengakibatkan kerugian finansial. Sebagai contoh apabila sistem yang diserang merupakan server yang menangani transaksi "commerce", maka apabila server tersebut tidak berfungsi, transaksi tidak dapat dilangsungkan. Misalnya dalam sebuah kasus perbankan. Sebuah Bank diserang oleh bank saingan dengan melumpuhkan outlet ATM (Anjungan Tunai Mandiri, Automatic Teller Machine) yang dimiliki oleh bank tersebut.

Meskipun DoS tidak membahayakan aplikasi dan data, namun berakibat aplikasi e-Layanan tidak dapat diakses. Untuk mengatasi DoS ini, cukup melakukan blok port yang ditarget oleh hacker.

2.3.2. Hacking

Hacking biasanya dilakukan oleh satu orang atau berkelompok. Orang yang melakukan tindakan hacking disebut hacker. Hacking adalah suatu usaha untuk menembus sistem dengan memanfaatkan kesalahan logika, konfigurasi ataupun mencoba-coba (brute force). Hacking terutama dilakukan di bidang jaringan komputer dengan menggunakan teknik tertentu. Hal yang dilakukan seorang hacker adalah mencari suatu celah dan kelemahan dari sebuah sistem,

baik itu sistem komputer maupun jaringannya. Dalam usaha mengamankan aplikasi web, perlu dilakukan pengecekan aplikasi secara berkala.

2.3.3. Phising

Phishing (Penyadapan atau plagiat) adalah aktivitas seseorang untuk mendapatkan informasi rahasia user dengan cara menggunakan email dan situs web yang menyerupai aslinya atau resmi. Informasi rahasia yang diminta biasanya berupa password account atau nomor kartu kredit, detail pembayaran, dan lain-lain. Kasus ini pernah menimpa situs BCA, dimana pelaku serangan membuat beberapa situs palsu dengan interface yang mirip dengan aslinya, namun dengan alamat yang berbeda. Pelanggan yang membuka situs BCA dialihkan kepada situs serupa tersebut. Dengan demikian, username dan password pelanggan tersebut dapat diketahui oleh pihak yang melakukan phising.

Target phishing adalah kecerobohan dan ketidaktelitian para pengguna jasa situs-situs jual-beli online, internet banking, online shopping dan sejenisnya yang melibatkan transaksi secara online melalui situs internet atau layanan telepon selular. Hal ini terjadi karena pengguna kurang waspada terhadap halaman yang mereka tuju.

2.3.4. SQL Injection

SQL injection adalah serangan yang memanfaatkan kelalaian dari website yang mengijinkan user untuk menginputkan data tertentu tanpa melakukan filter terhadap *malicious character*. Inputan tersebut biasanya di masukan pada box search atau bagian-bagian tertentu dari website yang berinteraksi dengan database SQL dari situs tersebut. Perintah yang dimasukkan para attacker biasanya adalah sebuah data yang mengandung link tertentu yang mengarahkan para korban ke website khusus yang digunakan para attacker untuk mengambil data pribadi korban.

Untuk menghindari link berbahaya dari website yang telah terinfeksi serangan SQL injection, dapat digunakan aplikasi tambahan seperti NoScript yang merupakan Add-ons untuk aplikasi web browser Firefox. Dan dewasa ini, sebagian besar teknologi web juga mampu menangani secara langsung dengan cara melakukan decode pada input atau parameter form tersebut.

2.3.5. Virus/Worm

Virus atau worm adalah agent yang dibuat oleh seseorang yang dengan tujuan tertentu untuk melakukan action-action yang merugikan host yang terinfeksi. Virus komputer adalah suatu program komputer yang menduplikasi atau menggandakan diri dengan menyisipkan kopian atau salinan dirinya ke dalam media penyimpanan/dokumen serta ke dalam jaringan secara diam-diam tanpa sepengetahuan pengguna komputer tersebut.

Worm adalah lubang keamanan atau celah kelemahan pada komputer yang memungkinkan komputer terinfeksi virus tanpa harus eksekusi suatu file yang umumnya terjadi pada jaringan.

Virus/worm sangat merugikan komputer jika terinfeksi. Efek dari virus komputer sangat beragam mulai dari hanya muncul pesan-pesan aneh hingga merusak komputer serta menghapus file atau dokumen. Hal terburuk yang bisa dilakukan virus atau work terhadap komputer yang terinfeksi adalah data pada komputer terhapus secara total dan permanen. Hal ini bisa dicegah dengan menginstall antivirus dan update definisi antivirus secara berkala untuk memastikan definisi antivirus terbaru.

2.4 Model dan Mekanisme Pengamanan

Pengamanan adalah proses yang mengikuti aturan tertentu. Aturan prosesnya didapatkan dari model pengamanan, dan aktivitas pengamanannya didefinisikan sebagai mekanisme pengamanan.

2.4.1. Otorisasi

Bentuk otorisasi yang diperbolehkan kepada pemakai (user) dalam suatu database dalam suatu perusahaan adalah sebagai berikut:

- Read authorization.

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembacaan data tetapi tidak diberikan ijin untuk melakukan modifikasi data yang ada.

- Insert authorization.

Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penyisipan data baru tetapi tidak diberikan ijin untuk melakukan modifikasi data yang ada.

- Update authorization.

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan modifikasi data tetapi tidak diberikan ijin untuk melakukan penghapusan data yang ada. Bentuk otorisasi yang diperbolehkan kepada pemakai (user) dalam memodifikasi skema database dalam suatu perusahaan adalah sebagai berikut :

- Indeks authorization.

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembuatan dan penghapusan indeks.

- Resources authorization

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan pembuatan relasi (hubungan) baru dalam database.

- Alteration authorization

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penambahan dan penghapusan atribut baru dalam relasi (hubungan) dalam database.

- Drop authorization

  Merupakan hak akses yang diperuntukkan user untuk diijinkan melakukan penghapusan relasi dalam database.

2.4.2. Model Role-Based Access Control (RBAC)

Dalam keamanan sistem komputer, kontrol akses berbasis peran (RBAC) ini adalah sebuah pendekatan untuk membatasi akses sistem untuk pengguna yang terotorisasi. RBAC ini juga dikenal dengan nama *Role-Based Security*.

Dalam suatu organisasi, peran (role) diciptakan untuk merepresentasikan berbagai fungsi pekerjaan. Hak akses untuk melakukan operasi/proses tertentu pada sistem diberikan kepada peran tertentu. Anggota staf (atau pengguna sistem lainnya) mendapatkan peran-peran tersebut, dan hanya melalui peran mereka mendapatkan izin untuk melakukan fungsi-fungsi sistem. Pengguna tidak berikan izin secara langsung untuk melakukan operasi tertentu, tetapi hanya mendapatkan

mereka melalui peran mereka, pengelolaan hak pengguna individu dipermudah dengan hanya menentukan peran yang sesuai untuk pemakai; hal ini menyederhanakan operasi umum, seperti menambah pengguna, atau mengubah unit organisasi si pengguna.

Ada tiga aturan utama dalam RBAC, yaitu:

1. **Pemberian Peran.** Seorang pengguna atau disebut juga subyek dapat menjalankan transaksi atau proses pada sistem hanya jika pengguna tersebut diberikan peran yang diizinkan untuk menjalankan transaksi tersebut. Peran bisa dianggap sebagai kelompok pengguna, seseorang diizinkan melaksanakan operasi yang menjadi hak kelompoknya. Pemberian peran dilakukan dengan menyatakan bahwa seorang pengguna adalah anggota dari kelompok tersebut.

2. **Otorisasi Peran.** Pemberian otorisasi kepada pengguna untuk menjalankan transaksi dilakukan dengan mengaktifkan peran yang hendak dipakai oleh pengguna tersebut. Dalam kondisi pengguna tidak melakukan apa-apa ke sistem, semua peran sifatnya pasif. Pengaktifan peran bersifat sementara selama sesi berlangsung. Apabila sesi berakhir, peran kembali di nonaktifkan. Dengan aturan nomor 1 di atas, dapat dipastikan bahwa peran yang diaktifkan hanya peran-peran yang memang dimiliki pengguna tersebut.

3. **Otorisasi Transaksi.** Pengguna dapat menjalankan transaksi hanya jika transaksi telah diotorisasi untuk peran pengguna yang diaktifkan. Dengan aturan nomor 1 dan nomor 2, dipastikan bahwa pengguna dapat melakukan transaksi hanya untuk yang mereka yang telah diotorisasi.

Untuk mendefinisikan model RBAC, beberapa konvensi berikut akan berguna:

- S = Subyek = Seorang pengguna atau aplikasi lain *(automated agent)*
- R = Peran = Fungsi pekerjaan atau jabatan yang menentukan tingkatan otoritas
- P = Permissions = Izin dari mode akses terhadap sumber daya
- SE = Sesi = Pemetaan yang melibatkan S, R dan/atau P
- SA = Pemberian peran ke subyek *(subject assignment)*
- PA = Pemberian izin akses ke proses/transaksi

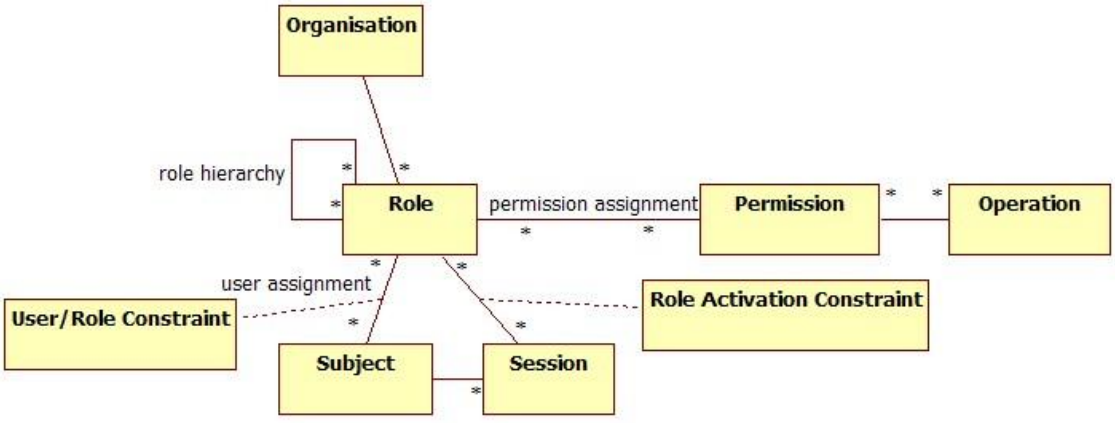
- RH = Hirarki peran yang urut parsial (partially ordered role hierarchy). RH juga dapat ditulis: ≥ (notasi ini: x ≥ y berarti x mewarisi hak akses y.)
- Subyek dapat memiliki peran lebih dari satu.
- Peran dapat mempunyai anggota beberapa subyek.
- Peran yang dapat memiliki lebih dari satu hak akses.
- Sebuah izin akses dapat diberikan ke lebih dari satu peran.

Pembatasan lain dapat dilakukan untuk mewujudkan aturan yang mengikat, misalnya untuk mencegah dua peran yang berlawanan dimiliki oleh seseorang. Contohnya peran untuk membuat account pengguna harus dipisah dengan peran untuk menentukan otorisasi aksesnya.

Dalam notasi himpunan, berlaku ketentuan sebagai berikut:

- PA $\subseteq$ P x R, adalah relasi many-to-many antara izin akses dan peran
- SA $\subseteq$ S x R, adalah relasi many-to-many antara subyek dan peran
- RH $\subseteq$ R x R

Kesemua konvensi di atas beserta dengan hubungannya dapat dilihat pada Gambar 1.



Gambar 1. Model Role-Based Access Control (RBAC).

2.4.3. Otentikasi

Otentikasi adalah adalah proses dalam rangka validasi user pada saat memasuki sistem. Nama dan password dari user dicek melalui proses yang mengecek langsung ke daftar mereka yang diberikan hak untuk memasuki sistem tersebut. Sifat mengetahui bahwa data yang diterima adalah sama dengan data yang dikirim dan bahwa pengirim yang mengklaim adalah benar-benar pengirim sebenarnya.

Metode Otentikasi

Otentikasi bertujuan untuk membuktikan siapa anda sebenarnya, apakah anda benar-benar orang yang anda klaim sebagai dia *(who you claim to be).* Ada banyak cara untuk membuktikan "siapa anda". Metode otentikasi bisa dilihat dalam 3 kategori metode:

1. ***Something You Know***

Ini adalah metode otentikasi yang paling umum. Cara ini mengandalkan kerahasiaan informasi, contohnya adalah password dan PIN. Cara ini berasumsi bahwa tidak ada seorangpun yang mengetahui rahasia itu kecuali anda seorang.

2. ***Something You Have***

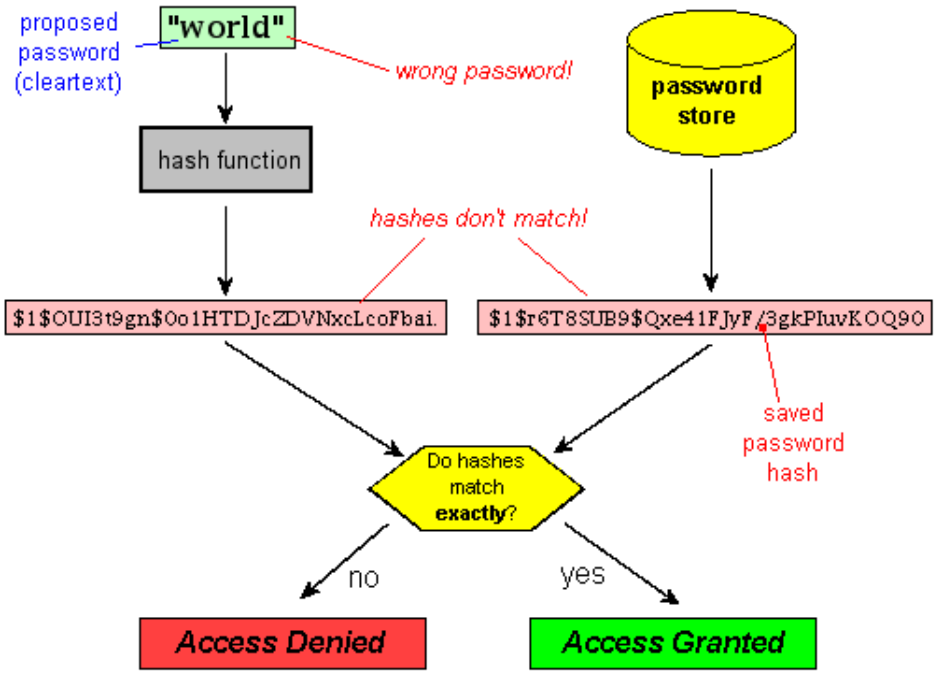
Cara ini biasanya merupakan faktor tambahan untuk membuat otentikasi menjadi lebih aman. Cara ini mengandalkan barang yang sifatnya unik contohnya adalah kartu magnetik/smartcard, hardware token, USB token dan sebagainya. Cara ini berasumsi bahwa tidak ada seorangpun yang memiliki barang tersebut kecuali anda seorang.

3. ***Something You Are***

Ini adalah metode yang paling jarang dipakai karena faktor teknologi dan manusia juga. Cara ini mengandalkan keunikan bagian-bagian tubuh anda yang tidak mungkin ada pada orang lain seperti sidik jari, suara atau sidik retina. Cara ini berasumsi bahwa bagian tubuh anda seperti sidik jari dan sidik retina, tidak mungkin sama dengan orang lain.

Proses Otentikasi

Seperti password pada umumnya, syarat agar otentikasi berhasil adalah password yang dikirimkan ke *client* = password yang disimpan di server. Dengan alasan keamanan jarang sekali server menyimpan password user dalam bentuk plain-text. Biasanya server menyimpan password user dalam bentuk hash sehingga tidak bisa dikembalikan dalam bentuk plain-text. Jadi syarat otentikasi berhasil di atas bisa diartikan sebagai hasil penghitungan hash dari password yang dikirim klien harus sama dengan nilai hash yang disimpan dalam server.



Gambar 2. Proses Otentikasi.

Penggunaan Salt

Untuk menghindari brute-force attack terhadap hash yang disimpan di server, maka sebelum password user dihitung nilai hashnya, terlebih dahulu ditambahkan string acak yang disebut dengan salt. Perhatikan contoh berikut, bila password user adalah “secret”, maka sebelum dihitung nilai hashnya, password ditambahkan dulu salt berupa string acak “81090273″ sehingga yang dihitung nilai hashnya adalah “secret81090273″ bukan “secret”.

Perhatikan bahwa nilai MD5 (“secret81090273″) adalah 894240dbe3d2b546c05a1a8e9e0df1bc sedangkan nilai MD5 (“secret”) adalah 5ebe2294ecd0e0f08eab7690d2a6ee69. Bila tanpa menggunakan salt, maka

attacker yang mendapatkan nilai hash 5ebe2294ecd0e0f08eab7690d2a6ee69 bisa menggunakan teknik brute force attack atau rainbow table untuk mendapatkan nilai password dalam plain-text. Salah satu contoh database MD5 online yang bisa dipakai untuk crack md5 adalah http://gdataonline.com/seekhash.php . Dalam situs tersebut coba masukkan nilai 5ebe2294ecd0e0f08eab7690d2a6ee69, maka situs tersebut akan memberikan hasil "secret". Hal ini disebabkan karena situs tersebut telah menyimpan pemetaan informasi secret<=>5ebe2294ecd0e0f08eab7690d2a6ee69.

Penambahan salt "81090273″ membuat nilai hash menjadi 894240dbe3d2b546c05a1a8e9e0df1bc. Bila nilai ini dimasukkan dalam situs tersebut, dijamin tidak akan ada dalam databasenya bahwa nilai hash tersebut adalah "secret81090273″. Karena nilai salt ini dibangkitkan secara random, maka tiap user memiliki nilai salt yang berbeda sehingga tidak mungkin attacker bisa membangun database pemetaan antara plaintext dan hash secara lengkap.

Dengan penggunaan salt, maka database pengguna dalam server akan tampak seperti tabel 2 berikut ini:

Tabel 2. Tabel Pengguna dengan Hasil Salting

Username	Salt	Password Hash
budi	81090273	894240dbe3d2b546c05a1a8e9e0df1bc

Field salt diperlukan ketika melakukan otentikasi. Password yang dikirimkan user akan ditambahkan dulu dengan nilai salt ini baru kemudian dihitung nilai hashnya. Nilai hash hasil perhitungan tersebut akan dibandingkan dengan field Password Hash yang ada di kolom sebelahnya. Bila sama, maka otentikasi berhasil, bila tidak sama, berarti otentikasi gagal. Secara prinsip sama saja dengan tabel 2, hanya ditambahkan satu langkah yaitu penambahan salt sebelum dihitung nilai hashnya.

BAB III
ANALISIS

Di sisi eksternal sistem e-Layanan merupakan sistem informasi yang merupakan kepanjangan tangan birokrasi untuk melayani masyarakat “dimana saja” dan “kapan saja”. Sedangkan di sisi internal, e-Layanan membantu birokrasi agar dapat bekerja efektif, efisien, akuntabel dan memberikan perlakuan yang adil bagi “siapa saja”. Kedua sisi tersebut membutuhkan partisipasi masyarakat, birokrat yang bertanggungjawab serta sistem yang sanggup mengamankan informasi dan proses sebagai hasil dari interaksi masyarakat – birokrat tersebut agar dapat mencapai tujuan sistem.

Tulang punggung sistem e-Layanan umumnya adalah aplikasi berbasis web. Aplikasi e-Layanan digunakan oleh masyarakat dan unit penyedianya. Karena melibatkan masyarakat luas, maka sistem e-Layanan harus aman terhadap ancaman eksternal. Demikian pula pemakaian internal oleh para pegawai unit penyedianya, sistem e-Layanan harus aman terhadap segala bentuk penyalahgunaan akses dari pihak internal yang tidak berwewenang. Efektivitas dan efisiensi yang hendak diraih dengan sistem e-Layanan tentunya juga akan menuntut bahwa pengelolaan dan pengaturan pengamanan e-Layanan harus mudah dan tidak kontra-produktif.

Dari kebutuhan pengamanan e-Layanan tersebut, ada beberapa faktor yang berpengaruh terhadap keamanan e-Layanan, yaitu:

1. Hal-hal yang perlu diamankan.
2. Ancaman-ancaman ketidakamanan dan penyalahgunaan
3. Cara mengatur dan mengendalikan keamanan.

3.1. Identifikasi Obyek-obyek dalam E-Layanan yang Diamankan

e-Layanan tidak lain adalah sistem informasi, karenanya elemen-elemen dalam e-Layanan adalah elemen sistem informasi, yaitu data dan proses. Agar e-Layanan dapat berjalan baik dan memberikan manfaat, elemen-elemen di dalamnya harus dijaga agar tidak mengalami perubahan di luar skenario sistem.

3.1.1. Identifikasi Elemen Data

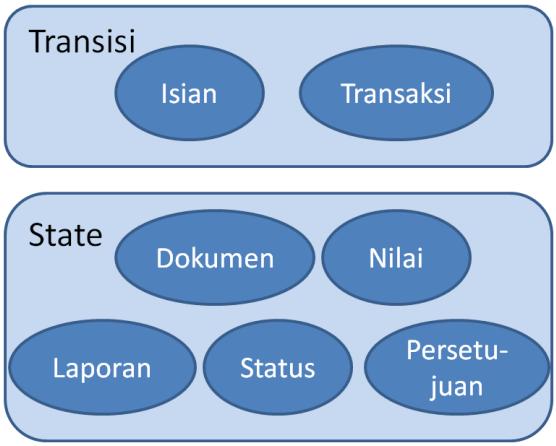
Data merupakan elemen terpenting dari suatu e-Layanan. Dengan adanya data, e-Layanan menjadi bermakna. Data sekaligus merupakan titik lemah keamanan suatu sistem. Hal ini tidak lain karena sifat data yang dinamis, selalu berubah. Backup ke

offline storage, seperti CD atau DVD tidak akan menyelesaikan masalah keamanannya, karena apa yang di-backup satu jam yang lalu mungkin saja menjadi tidak diperlukan lagi sekarang, karena ada yang lebih baru. Sedangkan backup yang dilakukan secara terus menerus juga tidak layak, karena kebutuhan kapasitasnya akan berkali-kali lipat lebih besar. Intinya, akan lebih sulit mengamankan sesuatu yang berubah daripada sesuatu yang statis.

Data yang berada dalam suatu e-Layanan dapat berupa teks dan angka-angka hasil isian, dokumen, gambar, laporan-laporan, hasil-hasil proses yang berupa status maupun persetujuan. Apapun yang memiliki makna dan dapat disimpan ke dalam *memory* komputer masuk dalam kategori data.

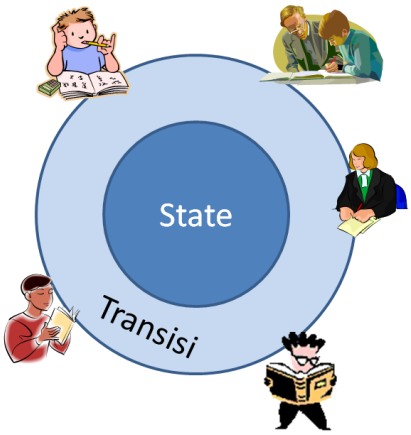
Secara umum, data yang bermacam-macam tersebut dapat dikategorikan menjadi *state* dan *transisi*. State merupakan teks, nilai maupun bentuk gambar dan dokumen yang menggambarkan suatu keadaan. Sedangkan transisi mewakili proses yang dapat mengubah keadaan tersebut. Singkatnya, transisi adalah catatan perubahan state. Transisi dan state dapat dianalogikan seperti proses dalam perbankan. Saldo tabungan merupakan state, sedangkan yang menjadi transisi adalah proses-proses yang dapat mengubah state, seperti melakukan setoran, pengambilan uang, dan transfer uang. Gambar 3 mengilustrasikan pengelompokan jenis-jenis data ke dalam state dan transisi.

Sudah merupakan hal yang lazim dalam suatu sistem informasi, state dapat dihitung berdasarkan kronologi dari transisinya. Jadi state merupakan data sekunder yang dapat direkonstruksi dengan menjalankan proses transisi yang menjadi data primer. Transisi merupakan rekaman penting yang menjelaskan kronologi perubahan state. Dengan menyimpan transisi, pertanyaan-pertanyaan "mengapa state itu bernilai sekian?" dapat terjawab.



Gambar 3. Pengkategorian Data e-Layanan dalam Kelompok Transisi dan State

Meskipun dapat dihitung, state tetap disimpan untuk alasan kepraktisan. Untuk menghindari state yang disimpan berbeda dengan hasil perhitungan kronologi transisi, biasanya pengisian atau pengubahan state tidak boleh dilakukan langsung secara manual. Semua perubahan state harus ada rekaman proses transisinya. Proses transisi yang disimpan itulah yang disebut transaksi. Integritas transaksi dan state dijaga dengan mengisolasi state dari akses pengguna secara langsung. Seperti halnya seorang teller bahkan manajer dari suatu bank tidak diperkenankan untuk mengubah saldo nasabahnya secara langsung. Seperti yang tergambar pada Gambar 4, semua perubahan state harus melewati transaksi.



Gambar 4. Model Pengamanan State Melalui Transisi.

Satu-satunya jalan untuk mengubah state adalah dengan memproses dan merekam transisinya dalam bentuk transaksi. Pengguna hanya boleh berinteraksi dengan proses transisi untuk dapat mengubah state, supaya setiap perubahannya tercatat.

State dalam hal ini merupakan data yang vital yang harus dilindungi lebih dari pengamanan transaksi. Perubahan state yang tidak melalui transaksi akan berpotensi menimbulkan kerugian.

3.1.2. Identifikasi Elemen Aplikasi

Bagian aplikasi yang berupa program, umumnya tidak sepenting data dalam hal kebutuhan pengamanannya. Apabila program rusak atau berubah, dengan mudah dapat dilakukan instalasi ulang dan tidak terlalu berdampak terhentinya layanan dalam waktu lama. Karena sifat itulah, maka program aplikasi dapat menjadi pelindung bagi elemen data. Pada sistem informasi yang operasional, seharusnya sudah tidak ada akses untuk

menambah, mengubah dan menghapus data langsung pada basisdatanya. Aplikasi harus menjadi satu-satunya jalan untuk melakukan perubahan data.

Dalam model pengamanan state melalui transisi yang telah dijelaskan di atas, transaksi merupakan proses yang diwadahi dalam aplikasi. Sehingga setiap form atau dialog yang merupakan realisasi proses pada apliasi perlu diamankan.

3.1.3. Identifikasi Tingkat Pengamanan Aplikasi e-Layanan

Tingkat kebutuhan pengamanan aplikasi e-Layanan berbeda menurut tingkat kompleksitasnya. Makin kompleks suatu e-Layanan, makin tinggi kebutuhan pengamanannya. Berikut adalah gambaran kebutuhan pengamanan dilihat dari tingkat kedewasaan e-Layanan:

1. Aplikasi hanya merupakan penyebar informasi *(information dissemination)* yang tidak mendapatkan input dari publik. Dalam hal ini, tidak ada celah yang memungkinkan publik untuk melakukan perubahan-perubahan dari luar sistem sebagai pengguna, sehingga model pengamanan state melalui transisi tidak diperlukan.

2. Aplikasi transaksional, dengan memberikan perlindungan pada data yang masuk kategori state agar tidak dapat diubah secara langsung melalui proses editing. State hanya dapat diubah melalui transaksi dimana setiap transaksi tersimpan dalam basis data. Apabila di kemudian hari ditemukan sesuatu yang salah pada state, dapat ditelusur balik melalui transaksinya. Pengamanan akses ke proses transaksi sudah memadai untuk aplikasi transaksional ini.

3. Aplikasi e-Layanan Terintegrasi Vertikal. Maksud dari integrasi vertikal ini adalah aplikasi yang berada pada domain yang sama tetapi digunakan oleh pengguna-pengguna di tingkat hirarki organisasi yang berbeda. Pengguna yang berada di induk organisasi tentunya butuh untuk dapat mengakses data lebih banyak daripada unit di bawahnya. Data yang dapat diakses seorang pengguna tergantung pada lingkup unit organisasi dimana dia berada. Data yang berada di luar lingkup organisasinya dicegah untuk diakses. Pengguna pada unit organisasi berbeda bisa sama-sama mendapatkan akses pada satu proses transaksi, akan tetapi dengan lingkup data yang berbeda.

4. Aplikasi e-Layanan Terintegrasi Horizontal. Maksud dari integrasi horizontal ini adalah aplikasi e-Layanan terhubung dengan aplikasi lain yang domainnya berbeda. Hubungan tersebut bisa berupa:

    a. *Data Sharing*. Aplikasi-aplikasi yang berbeda memakai basisdata yang sama.

    b. Interaksi antar aplikasi menggunakan *service* atau disebut juga *Service Oriented Architecture (SOA)*. Aplikasi yang satu "berbicara" dengan aplikasi lain secara langsung dengan mekanisme *web service* atau *remoting*.

Kebutuhan pengamanan untuk integrasi yang berupa data sharing umumnya tidak berbeda dengan e-Layanan yang transaksional atau yang masuk kategori integrasi vertikal. Akan tetapi, harus dipastikan bahwa semua aplikasi yang memakai data bersama tersebut mempunyai tingkat pengamanan yang sama. Apabila ada salah satu saja aplikasi yang menerapkan tingkat pengamanan yang lebih rendah, maka penerapan tingkat keamanan yang lebih tinggi pada aplikasi-aplikasi lainnya menjadi tidak berguna.

Turunnya tingkat keamanan karena adanya celah di salah satu aplikasi pada integrasi dengan model *data sharing*, dapat dilokalisir dengan penerapan *service oriented architecture*. Pada model integrasi menggunakan SOA, aplikasi lain yang meminta layanan diperlakukan seperti "seorang pengguna". Aplikasi yang mendapatkan layanan diberi hak akses pada proses transaksi di aplikasi yang melayani. Media interaksinya dapat berupa web service, atau remoting. Karenanya perlu pengamanan khusus pada jalur akses web service atau remoting tersebut, agar hanya aplikasi yang sah yang dapat menjalankan proses dan mencegah aplikasi-aplikasi liar untuk ikut-ikutan meminta layanan.

3.2. Identifikasi Ruang Lingkup Hak dan Wewenang bagi Pemangku Kepentingan

Berdasarkan model *Role-Based Access Control (RBAC)* dan kebutuhan akan integrasi aplikasi e-Layanan di Kemdiknas, dapat diidentifikasi faktor-faktor yang mempengaruhi hak dan wewenang bagi pengguna dalam berhubungan dengan e-Layanan, antara lain:

- **Proses/fungsi aplikasi**, yaitu pembatasan akses pengguna pada proses yang ada pada e-Layanan. Pengguna memperoleh hak akses akan proses e-Layanan sesuai dengan kewenangannya.

- **Ruang lingkup data**, yaitu pembatasan ruang lingkup data sesuai dengan wewenang unit organisasi tempat si pengguna berada.

- **Struktur organisasi pengguna sistem**, pengguna sistem dapat mengakses lingkup data pada unit organisasinya dan unit-unit organisasi yang ada dibawahnya.

Di samping ada tiga faktor yang menjadi pembeda hak-hak akses pengguna biasa pada sistem, ada kelompok/peran pengguna khusus yang perlu mendapat perhatian, yaitu *administrator*. Seorang administrator sistem mempunyai wewenang untuk memberikan dan mengatur akses pengguna lain. Karena wewenang ini, administrator dimungkinkan untuk menjadi celah keamanan dengan cara memberikan hak akses yang tinggi pada *account* seseorang, kemudian mencabutnya kembali setelah account tersebut dimanfaatkan untuk melakukan proses yang tidak sah.

3.3. Identifikasi Bentuk-Bentuk Ancaman dan Penyalahgunaan Hak Akses

Bagian ini akan menjelaskan tentang bentuk-bentuk ancaman dan penyalahgunaan hak akses yang berpotensi membahayakan sistem. Bentuk-bentuk ancaman dapat berupa banyak hal, baik dari internal maupun eksternal.

3.3.1. Identifikasi Bentuk-Bentuk Ancaman Keamanan Sistem

Bentuk-bentuk ancaman dapat berupa :

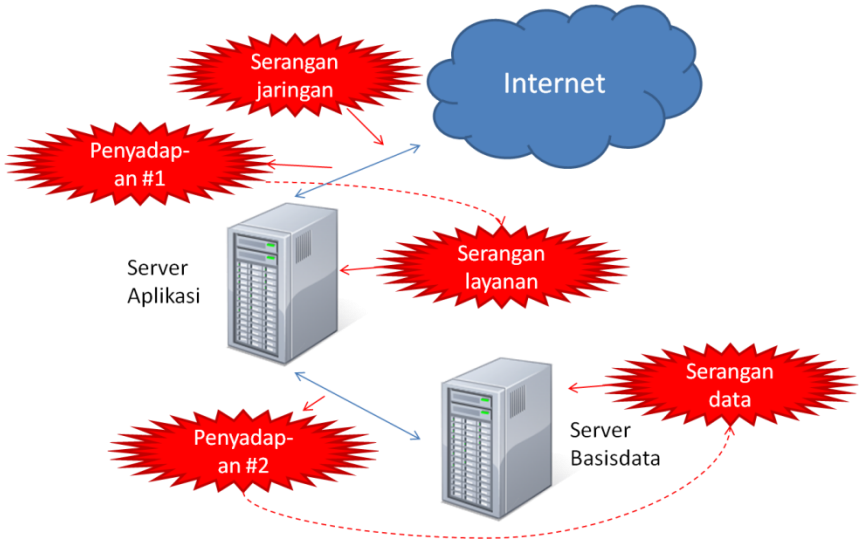
- Ancaman terhadap infrastruktur: Ancaman ini terdiri dari ancaman terhadap jaringan dan server.
- Ancaman terhadap Server Basis Data: Ancaman ini mengarah kepada data-data dari sebuah sistem.
- Ancaman terhadap Server Aplikasi: Ancaman ini lebih fokus kepada server aplikasi yang dibuat.

3.3.2. Pemetaan Bentuk-bentuk Ancaman ke Elemen-elemen Sistem

Potensi serangan atau ancaman keamanan dapat berasal dari luar sistem (eksternal) maupun dari para pemakai sistem (internal). Ancaman keamanan dari luar dapat dipetakan pada model deployment dari e-Layanan, sedangkan ancaman dari dalam dipetakan terhadap elemen aplikasi.

a. **Ancaman Eksternal**

Dari model deployment e-Layanan yang dapat dilihat pada Gambar 5, dapat dipetakan titik-titik rawan yang berpotensi menjadi sasaran serangan. Potensi ancaman dari luar siste dapat berupa perusakan maupun penyadapan. Memungkinkan juga merupakan kombinasi, yaitu penyadapan yang didahului serangan yang membuat sistem *malfunction*, kemudian pelaku memanfaatkan kelemahan sistem untuk mendapatkan data secara tidak sah. Adapun uraian penjelasan dari titik-titik rawan tersebut dapat dilihat pada Tabel 3.



Gambar 5. Model Ancaman Eksternal

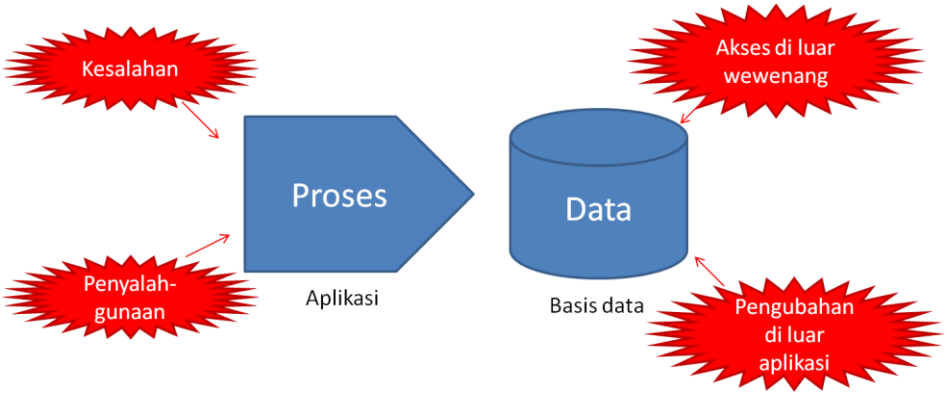
Secara umum, sasaran fisik dari ancaman eksternal adalah: jaringan, server aplikasi dan server basisdata. Dampak terburuk dari ancaman eksternal adalah server basisdata, yang mana perubahan-perubahan atau kerusakan pada server tersebut akan berakibat fatal. Sedangkan dampak paling ringan adalah apabila sistem kinerjanya terganggu, seperti misalnya melambat atau berhenti beroperasi untuk kurun waktu tertentu.

Tabel 3. Daftar Ancaman Eksternal

Ancaman	Sasaran	Pengamanan
Serangan jaringan	Jaringan	Firewall
Penyadapan #1	Data di tingkat Aplikasi	SSL
Serangan layanan (hijacking, SQL injection)	Akses ke aplikasi	Otentikasi
Penyadapan #2	Akses ke basisdata	Subnet terpisah, Enkripsi database connection string
Serangan basisdata	Data	Konfigurasi basisdata, Enkripsi password yang disimpan di basisdata

b. **Ancaman Internal**

Ancaman dari dalam sistem, terjadi apabila serangan eksternal berhasil menembus pertahanan e-Layanan. Ancaman internal ini juga dimungkinkan dilakukan oleh pihak-pihak di dalam Kemdiknas yang menginginkan perubahan-perubahan pada sistem secara tidak sah.



Gambar 6. Model Ancaman Internal

Secara umum, sasaran ancaman keamanan internal adalah proses dan data. Proses aplikasi dapat berupa form atau dialog isian, akan tetapi sasaran akhirnya adalah data. Adapun uraian dari ancaman terhadap proses dan data tersebut dapat dilihat pada Tabel 4.

Tabel 4. Model Ancaman Internal

Ancaman	Dampak	Pengamanan
Kesalahan	Data salah	Logging
Penyalahgunaan hak akses ke aplikasi	Kesalahan yang sistemik, umumnya merugikan	Otorisasi aplikasi dengan Role-based security, Logging
Akses di luar wewenang	Perubahan data di luar wewenang organisasinya	Otorisasi data dengan Record-level security, Logging
Pengubahan data di luar aplikasi	Perubahan data yang tidak tercatat di log,	Konfigurasi basisdata terpisah antara security dan data aplikasi,
	Pelanggaran integritas data	Enkripsi semua password yang tersimpan.

3.4. Kondisi Pengamanan e-Layanan Saat Ini

Dari hasil pengamatan, kondisi pengamanan e-Layanan yang ada saat ini masih mempunyai permasalah-permasalahan sebagai berikut :

- Masih banyak sistem e-Layanan yang belum mewadahi transaksi. Data dari penerima layanan dikirimkan melalui media dokumen, kemudian staf Kemdiknas akan melakukan ketik ulang dokumen tersebut untuk dimasukkan dalam basisdata. Memang kebutuhan pengamanan untuk e-Layanan yang masuk kategori ini tidak setinggi kebutuhan keamanan sistem transaksional. Tetapi manfaat yang didapat dari sistem seperti itu juga minimal.

- Pengamanan e-Layanan saat ini tidak terpusat sehingga setiap orang yang menginginkan e-Layanan yang berbeda akan mengulangi proses pendaftarannya (sign-up). Proses sign-up tersebut dilakukan sebanyak sistem e-Layanan yang dimasukinya.

BAB IV
MODEL KONSEPTUAL PENGAMANAN APLIKASI E-LAYANAN

Kondisi, permasalahan dan kebutuhan akan pengamanan pada Bab III menjadi dasar untuk perumusan solusinya. Rancangan konsepsual yang akan dibahas adalah merupakan rangkaian fungsi-fungsi yang harus ada dalam suatu manajemen pengamanan agar dapat menjawab permasalahan-permasalahan yang diidentifikasi pada Bab III.

4.1. Konsep Pengamanan (Security)

Tujuan suatu pengamanan untuk aplikasi e-Layanan tidak berbeda dengan tujuan pengamanan pada umumnya, yaitu mencegah akses yang tidak berhak ke sistem:

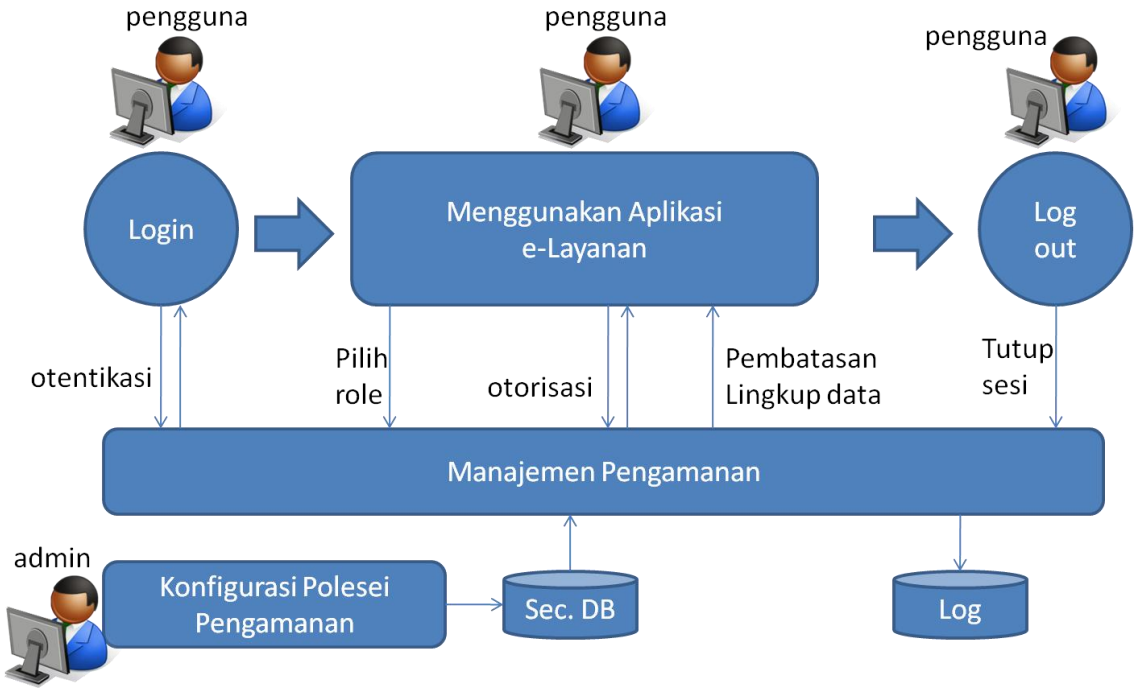
- Orang yang tidak berhak memakai sistem
- Pengguna yang berhak, tetapi haknya hanya untuk proses-proses tertentu saja, proses lain tidak berhak.
- Proses yang sedang dilakukan oleh pengguna, tetapi haknya hanya untuk lingkup data tertentu saja, data yang lain tidak berhak.

Untuk melakukan ketiga pembatasan akses di atas, masing-masing didekati dengan solusi:

- Otentikasi
- Role-Based Security (RBS)
- Row-Level Security (RLS) atau disebut juga Record Level Security.

4.1.1. Siklus Pengamanan E-Layanan

Siklus pengamanan e-Layanan mengikuti siklus penggunaannya. Akses ke sistem dimulai dengan otentikasi pengguna dengan cara melakukan *log in* ke sistem. Log in yang berhasil memungkinkan si pengguna melakukan proses-proses pada aplikasi. Akhir dari penggunaan aplikasi itu adalah pada saat pengguna *log out* atau disebut juga *log off* dari sistem. Setelah log out ini, pengguna sudah tidak dapat melakukan proses-proses pada sistem, kecuali bila pengguna log in lagi.



Gambar 7. Siklus Pengamanan.

Pengguna dapat melakukan login dengan sistem otentikasi yang sudah ada. Setelah melakukan login, pengguna dapat menggunakan aplikasi sesuai dengan otoritas masing-masing, dapat mengakses data sesuai dengan menunya. Setelah selesai melakukan aktifitas di dalam log in, pengguna dapat melakukan log out secara sengaja. Namun jika dalam jangka waktu tertentu (sesuai dengan pengaturan yang sudah ada), pengguna meninggalkan sistem dalam keadaan login, maka dengan otomatis sistem akan dikenai sesi tutup (logout). Selama ketiga aktifitas ini berlangsung, maka manajemen pengamanan terus berjalan. Admin melakukan konfigurasi-konfigurasi ke dalam database yang akan menjadi input dari manajemen pengamanan. Setiap pengguna yang sudah login dan logout akan tercatat secara otomatis dan terstruktur ke dalam sistem log.

4.1.2. Otentikasi

Otentikasi proses untuk memastikan siapakah si pengguna pada saat memulai pemakaian sistem. Ancaman terhadap otentikasi ini adalah penyadapan. Karenanya proses otentikasi harus dirancang agar tidak dapat ditiru oleh orang yang tidak berhak. Otentikasi terhadap sistem yang mempunyai kriteria sebagai berikut:

- Memastikan bahwa orang yang login adalah orang yang asli (pemilik login).

- Enkripsi asimetris. Ini dimaksudkan untuk menjaga agar karakter password lebih aman dan tidak diketahui oleh siapapun.

- Salting. Metode ini juga dimaksudkan untuk mengamankan password agar lebih bersifat unik dan rahasia.

- Apabila pengguna lupa passwordnya, password direset dengan pola yang acak dan pengiriman password yang direset tersebut melalui email. Hal ini ditujukan agar password lebih aman sampai ke tangan pemiliknya. Penyimpanan alamat email pada server juga harus dalam format terenkrip, agar tidak bisa dibelokkan ke email lain oleh orang yang berhasil membaca basisdata secara langsung.

Gambar 8. Otentikasi Sistem.

Pada Gambar 8 dapat dilihat proses melakukan login sesuai dengan otentikasi user. User asli memasukkan username dan password kemudian selanjunya diproses sistem sehingga dapat mengakses e-Layanan.

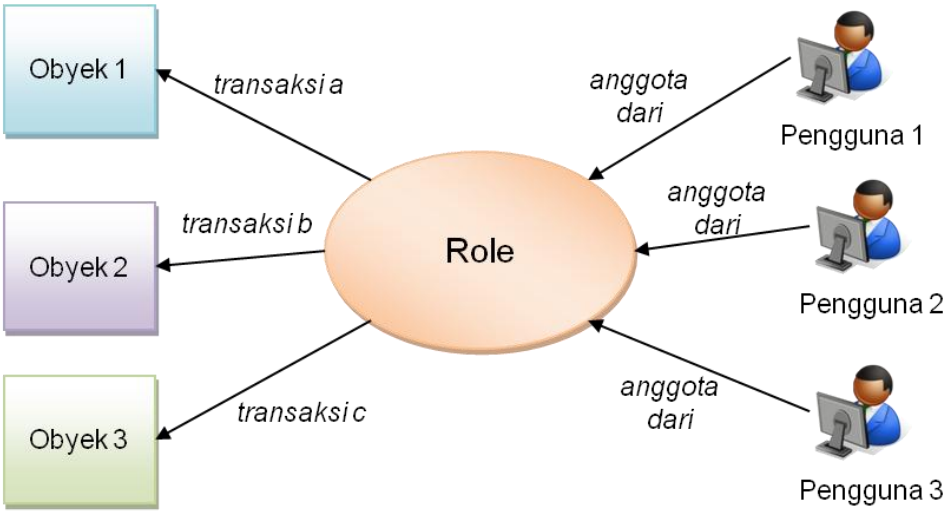
4.1.3. Otorisasi untuk Pembatasan Akses Proses

Pengguna yang berhasil masuk ke sistem tidak berarti bisa mengakses semua proses yang ada pada aplikasi. Ada pembatasan-pembatasan wewenang pengguna sebagai pelaku proses. Solusi terbaik yang ada pada saat ini untuk memberikan otorisasi pengguna akan proses-proses yang ada di aplikasi adalah penerapan polesei keamanan dengan model Role-Based Security.

4.1.3.1. Model Role-Based Security

Role-Based Security memberikan hak akses proses-proses ke pengguna melalui role atau peran. Pengguna tidak mendapatkan hak akses secara langsung ke proses-proses aplikasi, akan tetapi harus dikelompokkan dulu dalam role, barulah role tersebut diberi hak akses ke aplikasi. Hubungan tersebut dapat dilihat pada Gambar 9. Hal ini sejalan dengan pembagian wewenang di dunia nyata, bahwa tugas untuk melaksanakan proses/fungsi diberikan ke jabatannya. Siapapun yang sedang menduduki jabatan tersebut berhak dan berwewenang untuk melaksanakan prosesnya. Pola ini akan terasa

sekali manfaatnya apabila terjadi pergantian pejabat. Proses yang sedang ditangani tidak terus melekat ke orangnya apabila jabatannya telah diserahkan kepada penggantinya.



Gambar 9. Hubungan pengguna – *role* - proses.

Seorang pengguna juga dimungkinkan untuk menjadi anggota dari lebih dari satu role. Konsekuensi dari adanya multi-role ini, seorang pengguna akan mendapatkan hak akses pada proses-proses dari gabungan (union) dari masing-masing hak akses role yang ia miliki.

4.1.3.2. Pembatasan Sesi Pemakaian

Untuk e-Layanan yang berbasis web, pemberian hak akses untuk tiap form atau proses dilakukan per sesi pemakaian. Setiap kali pengguna yang melaksanakan suatu proses mendapat otorisasi, akan dibangkitkan satu kode untuk dipakai sebagai tiket pemakaian form tersebut. Kode tiket akan berbeda apabila pengguna mengakses form pada kesempatan yang lain. Dengan demikian, apabila kode tiket berhasil disadap, kode tersebut tidak berlaku lagi saat dipakai untuk melakukan serangan.

Sesi pemakaian suatu proses juga harus dibatasi waktunya. Apabila untuk beberapa saat tertentu pengguna tidak melakukan aktivitas apapun ke sistem dalam kondisi form yang masih terbuka, sesi pemakaian harus diakhiri secara otomatis (mekanisme *session time out*). Dengan pemberian sesi pada sistem tersebut, kecerobohan dan kelengahan user dapat direduksi dampaknya.

4.1.4. Pembatasan Akses Lingkup Data

Tiap pengguna hanya dapat melihat data yang menjadi kewenangannya. Meskipun pengguna memiliki menu dan bentuk form yang sama dengan pengguna lain di unit organisasi lain, ia menghadapi lingkup datanya sendiri yang berbeda dengan lingkup data unit organisasi lain. Lingkup data tersebut umumnya disesuaikan dengan hirarki/struktur organisasi.

Sesuai dengan model RBAC, bila seorang pengguna melaksanakan operasi pada sistem, ia terlebih dahulu mendapatkan sesi yang terotorisasi. Sesi inilah yang menentukan lingkup data yang dihadapi oleh pengguna tersebut. Apabila seorang pengguna mempunyai lebih dari satu peran, ia harus terlebih dahulu memilih peran yang diaktifkan. Hal ini dimaksudkan untuk menghindarkan kerancuan yang diakibatkan penggabungan lingkup data dari role yang berbeda. Jadi, pembatasan akses lingkup data hanya dimungkinkan diterapkan untuk satu role saja, yaitu role yang aktif.

4.2. Arsitektur

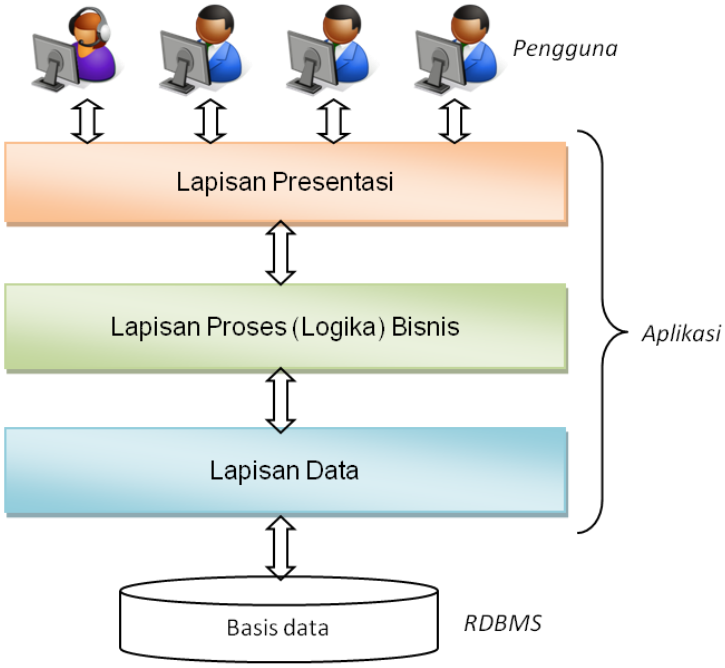
Arsitektur adalah cara untuk membagi sistem menjadi bagian-bagiannya dan menata hubungan antar bagian-bagian itu untuk mencapai tujuan sistem dengan pemenuhan keandalan, keamanan dan kinerja tertentu serta mampu mengantisipasi perubahan-perubahan di masa yang akan datang *(developability)*. Pembagian tersebut tidak hanya didasarkan pada pemenuhan fungsi sistem untuk menghasilkan informasi saja, tetapi pembagian tersebut menyederhanakan upaya-upaya untuk mencapai keandalan, keamanan, kinerja dan developability-nya.

Salah satu bentuk arsitektur yang menjadi best-practices dalam pengembangan perangkat lunak adalah arsitektur berlapis *(layered architecture)*. Perangkat lunak dibagi menjadi blok-blok yang tersusun vertikal membentuk lapisan-lapisan untuk memisahkan urusan tiap blok agar lingkupnya menjadi lebih sempit. Dengan mengecilnya lingkup blok ini, maka pembangunan, pengelolaan dan pemeliharaannya akan lebih mudah. Pembagian lapisan-lapisan yang lazim dipakai adalah sebagai berikut:

- **Lapisan presentasi.** Lapisan ini mengatur semua urusan interaksi antara pengguna dengan sistem. Tanggung jawab lapisan ini adalah untuk menampilkan informasi dan mendapatkan masukan dari pengguna. Pembangunan lapisan ini fokus pada upaya-upaya untuk membuat komunikasi antara pengguna dengan sistem berlangsung efektif dan bebas kesalahan.

- **Lapisan proses bisnis.** Implementasi dari aturan-aturan bisnis berada pada lapisan ini. Aturan-aturan tersebut dapat berupa: kapan suatu proses atau transaksi boleh dilakukan, oleh siapa, apa saja persyaratannya dan apa hasilnya. Semua yang sifatnya merupakan kendali akan proses atau transaksi ada pada lapisan ini.

- **Lapisan data.** Semua bahan untuk melakukan proses/transaksi maupun hasil-hasilnya adalah data. Lapisan ini bertanggungjawab untuk menyimpan, membaca, menghapus dan mengelola data untuk keperluan proses bisnis atau transaksi. Data biasanya disimpan dalam suatu basisdata yang saat ini banyak menggunakan *RDBMS (Relational Database Management System).* Lapisan ini bertugas untuk menangani hubungan sistem dengan basisdata.

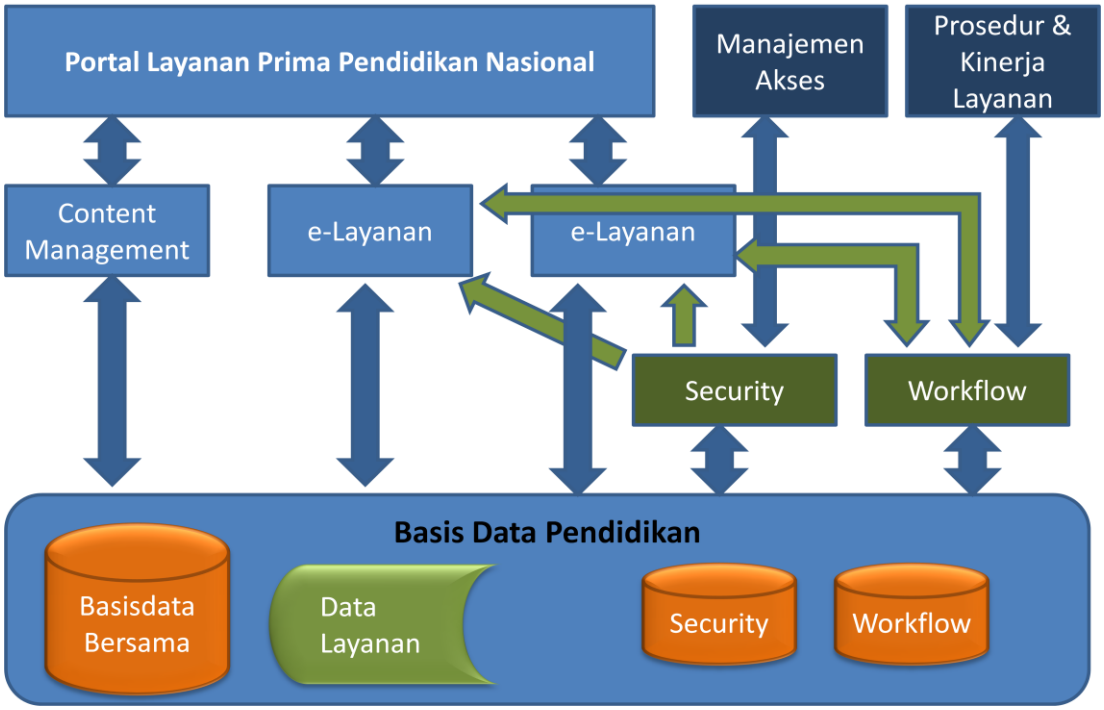


Gambar 10. Contoh arsitektur berlapis: "3-tier architecture".

Di luar ketiga lapisan tersebut masih dimungkinkan untuk menambahkan lapisan-lapisan lain sesuai dengan keperluan. Satu hal yang harus dijaga adalah pemisahan tanggung jawab setiap lapisan tersebut dilakukan secara ketat dan tertib. Lapisan proses, misalnya, tidak boleh mengandung sesuatu yang sifatnya untuk urusan tampilan, dan sebaliknya lapisan presentasi tidak sepatutnya mengandung rumus-rumus perhitungan suatu proses.

4.2.1. Arsitektur Sistem-Sistem di Kementerian

Arsitektur berlapis juga dapat dipakai untuk menggambarkan hubungan antar sistem dalam satu organisasi. Gambar 11 menunjukkan pedoman arsitektur sistem-sistem yang dikeluarkan oleh Tim Reformasi Birokrasi Internal (RBI) Kemdiknas. Masyarakat mendapatkan informasi serta layanan melalui satu pintu utama, yaitu Portal Layanan Prima Pendidikan Nasional. Portal layanan prima tersusun atas content management dan e-Layanan dan semua e-Layanan berada dalam lapisan di bawah Portal tersebut.



Gambar 11. Panduan RBI tentang Arsitektur.

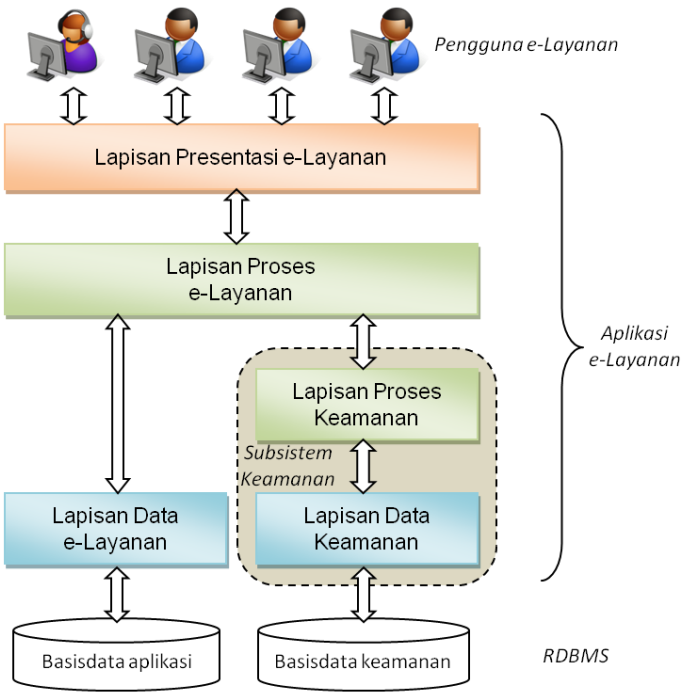
Pengamanan (security) adalah termasuk menjadi bagian dari arsitektur. e-Layanan dibuat dengan menggunakan workflow yang bagus dan pengamanan yang kuat, baik dari sisi data, aplikasi, maupun workflownya yang terangkum dalam basis data pendidikan.

Untuk melakukan pengelolaan pengamanan, dibuat aplikasi khusus yaitu "Manajemen Akses" yang hanya dapat diakses oleh pada administrator sistem. Manajemen Akses bukan merupakan aplikasi publik, melainkan aplikasi tertutup yang pengoperasiannya membutuhkan otorisasi khusus.

4.2.2. Arsitektur e-Layanan

Dengan mengikuti pedoman arsitektur sistem di atas, maka arsitektur e-Layanan harus mempunyai lapisan keamanan didalamnya. Basisdata untuk keamanan dipisah dengan basisdata aplikasi agar tidak terjadi lubang keamanan yang diakibatkan oleh administrator aplikasi. Hal ini mengingat administrator aplikasi memungkinkan mempunyai akses langsung pada basisdata aplikasinya. Gambar 12 menjelaskan arsitektur e-Layanan tersebut.

Dalam Gambar 12, Subsistem Keamanan terdiri atas dua lapisan, yaitu Lapisan Proses Keamanan dan Lapisan Data Keamanan. Hal ini dikarenakan dukungan pengamanan untuk aplikasi terdiri atas proses-proses dan mengacu pada data konfigurasi keamanan. Lapisan Proses Keamanan yang berada dalam aplikasi disebut juga sebagai *Security Agent*.



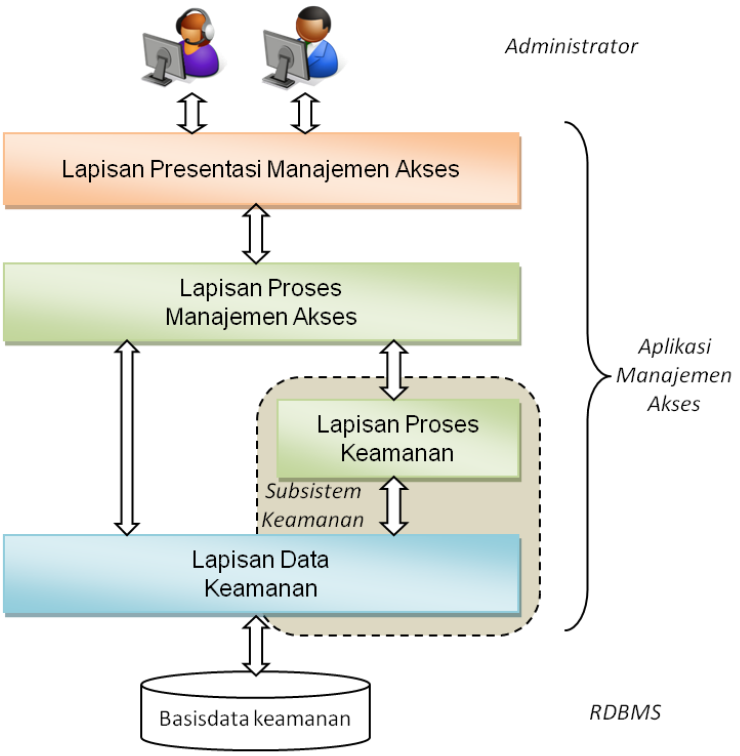
Gambar 12. Arsitektur e-Layanan yang memasukkan Subsistem Keamanan.

4.2.3. Arsitektur Manajemen Akses

Seperti yang telah dijelaskan sebelumnya bahwa Manajemen Akses adalah aplikasi untuk administrator. Manajemen Akses juga terikat dengan polesei keamanan yang sama dengan e-Layanan. Karenanya Lapisan Proses Keamanan juga harus diterapkan pada arsitektur Manajemen Akses. Adapun arsitektur untuk Manajemen Akses ini dapat dilihat pada Gambar 13.

Fungsi-fungsi Manajemen Akses adalah kumpulan transaksi atau pengelolaan yang dibutuhkan untuk melakukan konfigurasi sesuai dengan ketentuan RBAC dan Row Level Security, yaitu:

- Pengelolaan subyek atau pengguna
- Pengelolaan jenis transaksi/proses/operasi
- Pengelolaan role
- Pengelolaan pemberian role kepada pengguna
- Pengelolaan pemberian hak transaksi kepada role
- Pengelolaan organisasi
- Pengelolaan posisi organisasi pengguna
- Penentuan otorisasi operasi baca, tulis, ubah dan hapus untuk lingkup data sub unit organisasi di bawah unit masing-masing pengguna
- Mendelegasikan hak administrasi ke administrator di bawahnya
- Melihat log operasi/log sesi.



Gambar 13. Arsitektur Manajemen Akses.

4.2.4. Security Agent

Security Agent adalah lapisan proses keamanan yang mengatur keamanan sistem. Agent ini memverifikasi hak akses pengguna, sesi pengguna, mencatat log operasi. Hal ini dapat diilustrasikan seperti berikut: Setelah pengguna login ke sistem, Security Agent akan mengecek kevalidan akun (ID) pengguna dan password. Jika valid, maka Security Agent akan memberikan hak akses sesuai dengan role yang dipegang oleh pengguna tersebut. Untuk setiap operasi yang dilakukan oleh pengguna yang berdampak pada perubahan basisdata, Security Agent dapat mencatatnya dalam log operasi. Log ini nantinya dibutuhkan apabila diperlukan telusur balik untuk melihat kronologi perubahan data.

Security agent harus dipasang pada semua form/dialog transaksi maupun laporan. Otorisasi memang bisa dipakai untuk mengatur pemunculan menu. Menu-menu yang tidak menjadi hak si pengguna untuk mengaksesnya akan disembunyikan. Akan tetapi untuk aplikasi web, hal itu tidak cukup. URL dari form atau laporan bisa disimpan/di-bookmark oleh si pengguna dan pengguna di kemudian hari dapat menuju form atau laporan tersebut tanpa melalui menu aplikasi. Hal ini merupakan celah keamanan, karenanya pada setiap form/laporan perlu dicegah untuk akses semacam ini dengan pemasangan security agent. Dengan demikian setiap akses ke form/laporan, sesi yang dihasilkan dari otorisasi terhadap role yang aktif diverfikasi ulang.

Penolakan atau *access violation* yang terdeteksi oleh security agent, harus memaksa aplikasi untuk pindah ke halaman lain. Halaman lain tersebut biasanya berupa pesan peringatan, bahwa akses tidak diperbolehkan dengan menyebutkan penyebabnya.

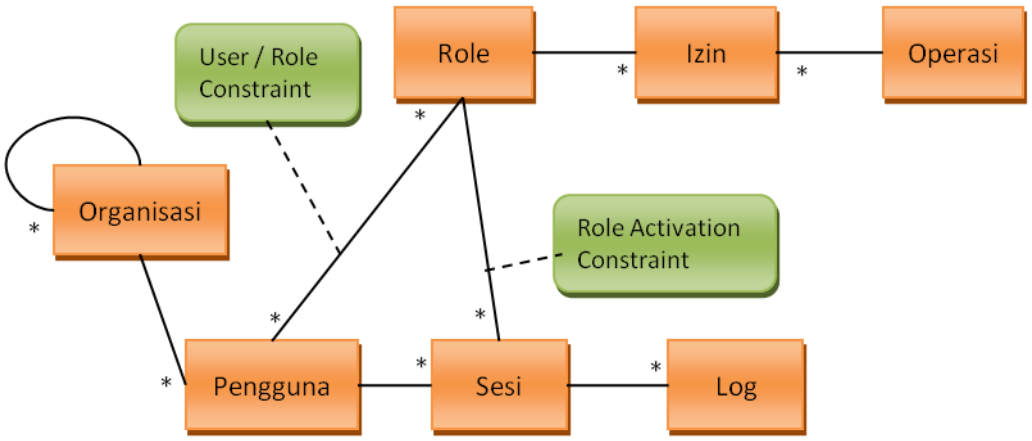
4.3. Model Pengamanan Aplikasi e-Layanan

Dalam setiap aplikasi e-Layanan, dukungan keamanannya harus sekaligus mengimplementasikan *Role-Based Access Control (RBAC)* dan *Row Level Security (RLS)* sekaligus. Kedua model pengamanan ini akan menjadi gerbang pengendali akses ke aplikasi sekaligus juga pengendali akses ke datanya.

Mengingat besarnya organisasi Kemdiknas, dimana peran unit organisasi dominan dalam menentukan lingkup tanggung jawab seorang pengguna, maka perlu sedikit modifikasi dari definisi struktur RBAC yang asli. Perubahan tersebut adalah pada pergeseran penentuan hirarki role (RH) menjadi hirarki organisasi (OH). Adapun hasil perubahan tersebut dapat dilihat pada Gambar 14.

Dengan memindahkan hubungan antara organisasi dan role, memungkinkan untuk memakai role yang sama untuk lingkup unit organisasi yang berbeda. Hal ini

akan menyederhanakan definisi model pengamanan untuk organisasi yang hirarkinya terdiri atas banyak tingkat seperti Kemdiknas.



Gambar 14. Modifikasi struktur RBAC untuk Kemdiknas.

Dengan struktur role yang berada di dalam organisasi tersebut, sekaligus juga mengakomodir mekanisme *Row Level Security*. Setiap record data pada aplikasi mendapatkan atribut hirarki organisasi (OH) dari pengguna yang membuatnya *(record creation)*. Dengan berdasar informasi ID unit organisasi yang ada pada masing-masing record inilah dapat diketahui lingkup organisasi dari record tersebut. Apabila sesudahnya ada pengguna lain yang hendak mengakses record tersebut, sesi pengguna baru tersebut akan dievaluasi terhadap atribut organisasi si pencipta record. Apabila $OH_{pengguna} \geq OH_{record}$, maka si pengguna baru mendapat otorisasi untuk operasi terhadap record tersebut. Sebaliknya bila $OH_{pengguna} < OH_{record}$, si pengguna tidak dapat mengaksesnya sama sekali.

4.3.1. Pencatatan Log *(Logging)*

Pada Gambar 14 juga dapat dilihat posisi log yang dipakai untuk mencatat perubahan-perubahan yang dilakukan pengguna selama sesi berlangsung. Dalam satu sesi dapat menghasilkan lebih dari satu record di log. Dengan demikian, pengguna yang berusaha untuk “menghilangkan jejak” dengan menghapus record, masih bisa diketahui operasi penghapusannya melalui log ini.

4.3.2. Mekanisme Time Out

Apabila pengguna dalam rentang waktu tertentu tidak melakukan aktifitas ke e-Layanan, maka akan secara otomatis di Logout oleh Security Agent. Mekanisme ini sangat membantu pengguna untuk menghindari penyalahgunaan oleh pengguna lain yang tidak berhak jika pengguna aslinya meninggalkan aplikasi dalam kondisi terbuka.

4.3.3. Verifikasi Pengguna Baru dan Pemberian Password

Dalam hal si pengguna baru sudah dikenal oleh administrator, maka hak akses bisa langsung diberikan oleh administrator melalui aplikasi Manajemen Akses. Hal ini berlaku bila yang menjadi pengguna adalah pegawai Kemdiknas saja (aplikasi internal).

Khusus untuk aplikasi publik, yang penggunanya bisa berbagai kalangan dimana tidak mungkin administrator dapat mengenalnya satu persatu, maka pengguna baru tersebut perlu melalui proses verifikasi. Proses ini dimaksudkan agar transaksi yang dihasilkan oleh pengguna tersebut nantinya sah dan bersifat mengikat. Si pengguna tidak dapat memungkiri setiap transaksi yang telah ia lakukan *(prinsip non-repudiation).*

Tahapan dalam melakukan verifikasi terhadap pengguna baru adalah sebagai berikut:

- Pengguna mendaftarkan diri pada sistem dengan menyebutkan identitas lengkap beserta alamat e-mail-nya. Untuk menghindari pendaftaran otomatis yang dilakukan oleh *web robot*, form pendaftaran ini sebaiknya menggunakan *captcha*. Penjelasan tentang captcha dapat dilihat pada seksi 4.3.4.

- Alamat e-mail diverifikasi apakah sudah pernah terdaftar atau belum. Apabila sudah terdaftar, pengguna ditawari untuk mereset passwordnya dan proses verifikasi tidak dilanjutkan.

- Proses verifikasi pengguna baru akan dilakukan oleh sistem dengan cara mengirimkan kode aktivasi ke e-mail pengguna yang telah mendaftar tersebut.

- Pengguna harus melakukan konfirmasi dengan melakukan registrasi kode aktivasi tersebut dari e-mail-nya. Pengguna cukup melakukan *single-click* hyperlink yang ada di e-mail-nya untuk melakukan konfirmasi, tidak perlu melakukan menginputkan/mengetik sesuatu. Dengan langkah ini, pengguna telah dijamin memang benar-benar pemilik e-mail tersebut. Atau dengan kata lain, pengguna dan e-mailnya berkorespondensi satu-satu.

- Apabila ada permintaan untuk pengubahan password, konfirmasi untuk pengubahan password dikirimkan kembali melalui e-mail yang telah terdaftar. Apabila si pengguna menyetujui pengubahan password, barulah password di-reset (di-generate ulang secara acak) dan password barunya

dikirimkan melalui e-mail yang sama. Dengan cara ini pengguna bebas dari ancaman pengubahan password oleh orang yang tidak berhak.

4.3.4. Penggunaan Captcha

Captcha[1] adalah suatu bentuk pengujian *(challange-response test)* yang digunakan aplikasi web untuk memastikan bahwa pengguna adalah seorang manusia dengan cara meminta jawaban yang hampir tidak mungkin dihasilkan oleh komputer. Proses ini biasanya dilakukan server yang meminta seorang pengguna untuk menyelesaikan suatu pertanyaan sederhana yang dapat dihasilkan dan dinilai oleh komputer tersebut. Bentuk pertanyaan tersebut sederhana bagi manusia, tetapi tidak sederhana untuk membuat program yang dapat menjawab pertanyaan itu. Karenanya, pengguna yang dapat memberikan jawaban yang benar akan dianggap sebagai manusia.

Gambar 15. Contoh Captcha.

Captcha umumnya menggunakan huruf dan angka dari citra terdistorsi yang muncul di layar. Pengguna diminta mengetik captcha tersebut ke kolom isian. Apabila hasil evaluasi cocok dengan kata aslinya, maka pengguna diperbolehkan untuk meneruskan proses pendaftarannya. Captcha bisa juga merupakan pertanyaan hitungan aritmetika sederhana, tetapi semua angkanya disebutkan dengan kata-kata (memakai huruf). Beberapa contoh captcha dapat dilihat pada Gambar 15.

4.3.5. Pendelegasian Administrator

Kemdiknas mempunyai unit kerja yang sangat banyak. Tidak mungkin menangani semua pengguna yang tersebar di seluruh Indonesia secara terpusat, karena tidak mungkin tim administrator yang di pusat mengetahui kebenaran setiap penggunanya. Karenanya, perlu mendelegasikan manajemen akses ke administrator ke

[1] Wikipedia, http://en.wikipedia.org, key=”web service”, 2010.

unit-unit kerja di bawahnya. Di tingkat unit kerja, bisa saja terjadi hal yang sama yang mengharuskan mendelegasikan lagi administrasi pengguna pada administrator sub-unitnya.

Mekanisme pendelegasian administrator adalah sebagai berikut:

- Secara umum Administrator adalah sebuah role. Role tersebut akan didelegasikan kepada pengguna yang bertindak sebagai administrator.
- Pengguna yang ditunjuk menjadi administrator yang menerima delegasi adalah pengguna yang ada di unit organisasi di bawah unit organisasi administrator pemberi delegasi.
- Administrator pemberi delegasi membuatkan role administrator di unit organisasi di bawahnya.
- Administrator menyebutkan pengguna-pengguna di unit organisasi di bawahnya untuk dimasukkan dalam role administrator yang dibuatnya tadi.
- Untuk alasan akuntabilitas, sebaiknya pemberikan hak akses ke transaksi ke role administrator yang baru hanya boleh diberikan oleh administrator yang tidak menunjuk pengguna.

4.3.6. Pengamanan Web Service

Web service merupakan media untuk mekanisme integrasi antar aplikasi yang paling ampuh saat ini. Web service menyederhanakan penerapan *Service Oriented Architecture (SOA)* untuk membuat aplikasi yang satu mampu berbicara dengan aplikasi yang lain, seperti halnya seorang pengguna berbicara kepada aplikasi itu.

Web service ini juga dapat mengisolasi akses ke database aplikasi lain. Dengan adanya basisdata yang terisolir ini, tidak perlu ada *exposure* basisdata ke internet maupun intranet yang berpotensi untuk mendapatkan serangan. Hacking terhadap web service akan menimbulkan dampak yang jauh lebih kecil dibandingkan dengan hacking terhadap basisdata. Semua operasi/transaksi yang dilakukan melalui web service dapat dicatat pada log. Karenanya apabila ada penyalahgunaan web service, akan mudah untuk diketahui.

Akses aplikasi lain melalui web service tidak ubahnya seperti pengguna manusia. Aplikasi lain yang hendak melakukan sesuatu harus melakukan otentikasi dahulu. Kemudian dari otentikasi tersebut, aplikasi mendapatkan tiket atau token untuk membuka sesi untuk melakukan operasi. Tiket tersebut bersifat dinamis. Apabila sesi sudah diakhiri, tiket tersebut tidak berlaku lagi untuk sesi yang lain.

Apabila tiket dapat terendus oleh pihak penyerang, maka penyerang hanya punya kesempatan untuk memakainya selama sesi berlangsung. Apabila sesi berlangsung sangat singkat, maka tidak ada lagi kesempatan bagi penyerang untuk melancarkan serangan. Karenanya, siklus open – close sesi ini harus berlangsung singkat. Satu operasi harus satu sesi saja.

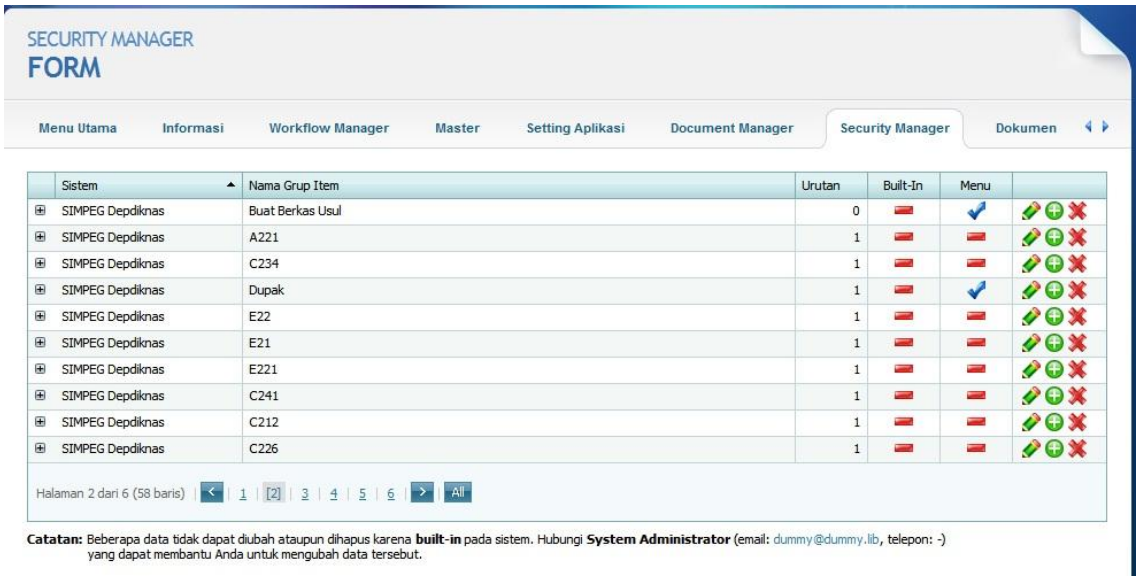
Enkripsi dengan *Secure Socket Layer (SSL)* sebaiknya diterapkan untuk web service yang sifatnya sensitif atau bernilai tinggi. Enkripsi dengan SSL ini akan mempersulit penyerang untuk mendapatkan profil web service yang dibuka.

4.4. Rancangan Antar Muka Manajemen Akses

Aplikasi Manajemen Akses adalah aplikasi pendukung bagi e-Layanan. Fungsi aplikasi ini adalah untuk mengelola dan mengatur akses pengguna ke sistem. Penempatan aplikasi ini disatukan ke dalam aplikasi e-Layanan maupun berdiri sendiri. Pengguna aplikasi ini hanyalah administrator yang mendapatkan wewenang untuk pengaturan akses. Rancangan fungsional berikut ini terdiri atas fitur-fitur yang harus ada dalam aplikasi Manajemen Akses.

4.4.1. Pengaturan Elemen Aplikasi

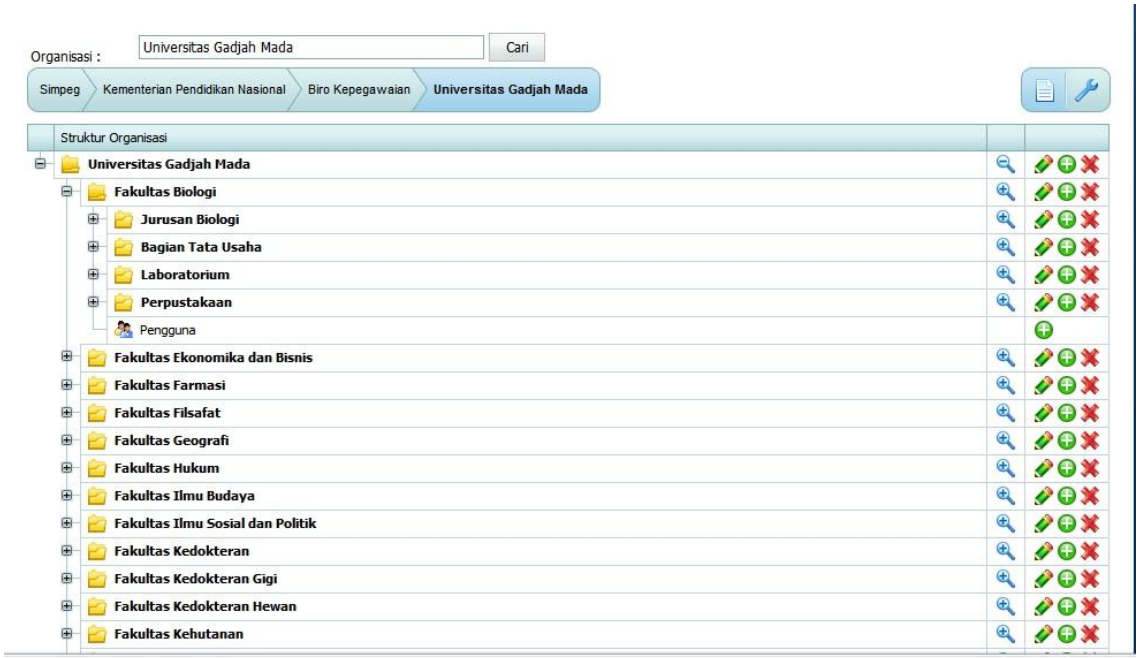
Elemen aplikasi dapat berupa form transaksi, laporan atau apapun yang dapat diakses melalui menu aplikasi. Menu aplikasi untuk kemudahan penataan, kadang-kadang perlu pengelompokan, yang disebut dengan grup menu. Agar pengelolaan sederhana, grup menu ini termasuk dalam item yang diatur sebagai elemen aplikasi. Sedangkan satu sistem dapat terdiri atas beberapa elemen aplikasi. Rancangan antar muka untuk pengelolaan elemen aplikasi dapat dilihat pada Gambar 16.



Gambar 16. Pengaturan elemen aplikasi.

4.4.2. Pengaturan Organisasi dan Pengguna

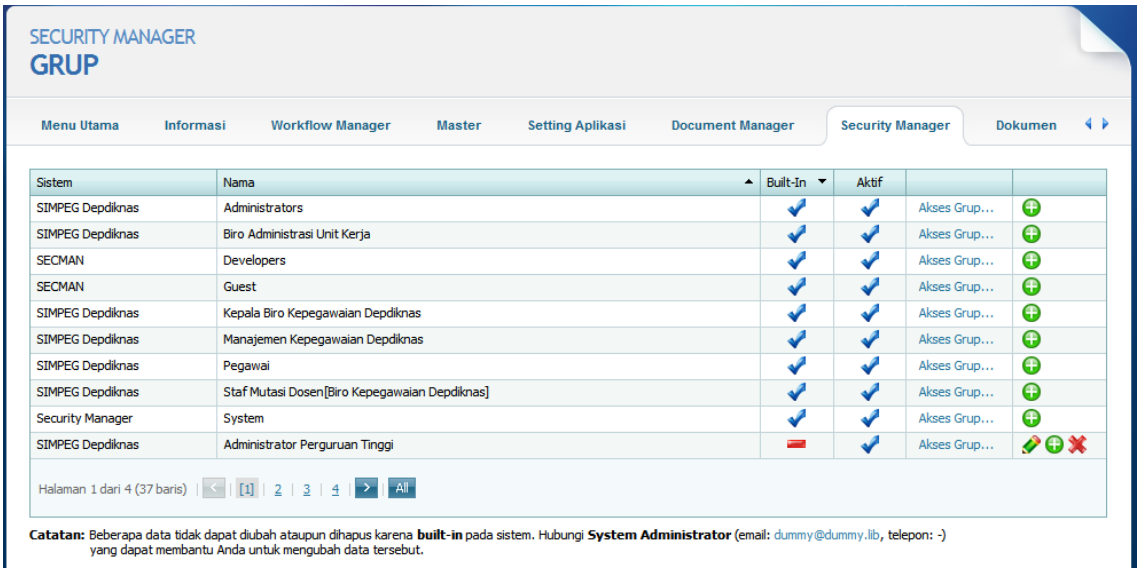
Gambar 17 merupakan rancangan antar muka untuk pengaturan organisasi seperti Institut, universitas, fakultas, jurusan, hingga sampai pada level organisasi terkecil seperti laboratorium. Dalam rancangan ini pengelolaan organisasi sekaligus juga merupakan pengelolaan pengguna. Hal ini dikarenakan definisi unit organisasi di Kemdiknas cukup dominan. Pengguna diletakkan dalam unit organisasi dalam grup "Pengguna". Pengguna dapat dipindahkan ke unit organisasi yang lain.



Gambar 17. Pengaturan organisasi dan pengguna.

4.4.3. Pengaturan Role

Role, sesuai dengan deskripsi pada Gambar 14, tidak mempunyai hirarki seperti versi asli RBAC. Hirarki yang mempnegaruhi lingkup data sudah diakomodir di struktur organisasi. Melalui fitur Pengaturan Role ini, jabatan-jabatan maupun peran yang sifatnya non jabatan dapat dibuat sebagai "Grup Pengguna". Gambar 18 merupakan rancangan antar muka untuk pengaturan role/kelompok pengguna. Izin akses ke transaksi atau elemen aplikasi dapat dilakukan melalui form ini dengan mengklik link "Akses Grup".



Gambar 18. Pengaturan role/kelompok pengguna.

4.4.4. Izin Akses Role ke Elemen Aplikasi

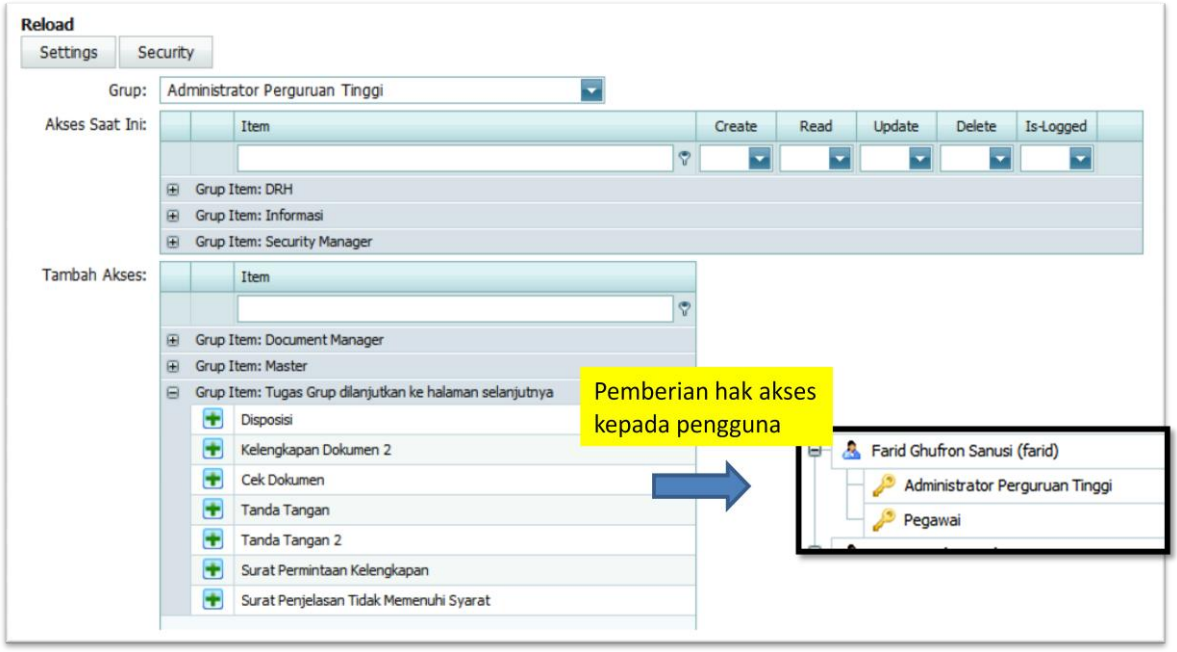
Izin akses roke ke elemen aplikasi ini merupakan ekstensi dari Pengaturan Role. Grup pengguna atau role yang dipilih akan mengubah tampilan detilnya, yaitu izin aksesnya ke elemen aplikasi. Satu role dapat mempunyai izin ke banyak elemen aplikasi. Pengaturan izin akses role ini harus memungkinkan untuk menambah atau mengurangi izin akses tersebut secara mudah dan bebas kesalahan ketik.

Rancangan antar muka untuk hak akses role ke elemen aplikasi dapat dilihat pada Gambar 19. Hak akses diberikan kepada role atau grup pengguna dengan cara klik satu tombol saja. Melalui fitur ini pula, opsi jenis hak akses ke datanya dapat didefinisikan, yaitu:

- Membuat record baru
- Membaca

- Mengubah
- Menghapus
- Mencatat semua transaksinya ke log.

Karena log membutuhkan memory / penyimpanan yang sangat besar, maka penggunaannya harus selektif. Untuk itu perlu disediakan opsi untuk mencatat aktivitas role ke elemen aplikasi atau tidak. Melalui fitur ini, opsi tersebut dapat dipilih untuk diaktifkan atau tidak.

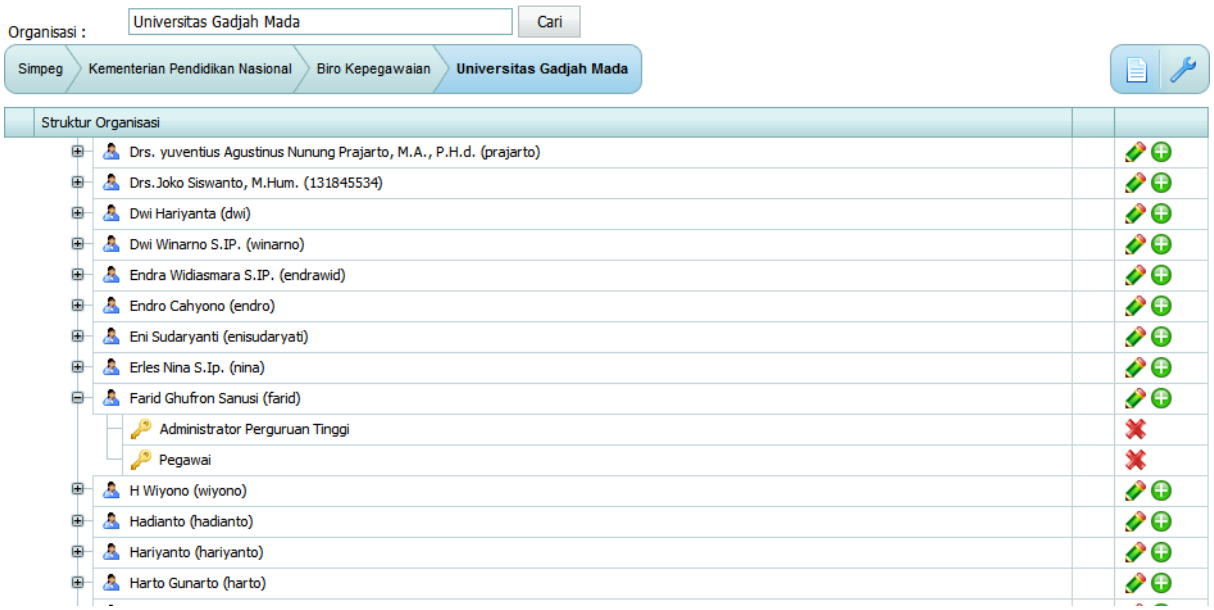


Gambar 19. Pengaturan Izin akses role ke elemen aplikasi.

4.4.5. Pemberian Role ke Pengguna

Pemberian role ke pengguna *(role assignment)*, harus dapat dilihat dari sudut pandang penggunanya maupun dari sudut pandang role-nya. Sudut pandang pengguna, maksudnya adalah untuk setiap pengguna, administrator dengan mudah dapat melihat pengguna tersebut mempunyai role apa saja. Sebaliknya dari sudut pandang role, administrator dapat melihat siapa saja pengguna-pengguna yang tergabung dalam suatu role. Gambar 19 merupakan rancangan antar muka untuk pengaturan pengguna dari sudut pandang pengguna.

Pengelolaan pemberian role ke pengguna ini harus mewadahi pemberian role secara mudah dan membebaskan pengguna dari kesalahan ketik.



Gambar 20. Pengaturan pemberian role kepada pengguna.

4.5. Security sebagai Soft Infrastruktur

Pengamanan e-Layanan, apapun aplikasinya dapat dipenuhi dengan RBAC dan RLS. Semua kebutuhan manajemen pengamanannya mempunyai pola yang sama. Selalu ada cara untuk merepresentasikan berbagai kebutuhan pengamanan ke dalam bentuk model hasil modifikasi penggabungan RBAC dan RLS. Karenanya model manajemen pengamanan di atas dapat diterapkan *(reusable)* untuk berbagai e-Layanan maupun aplikasi internal.

Pemakaian manajemen pengamanan ini untuk berbagai e-Layanan secara terpusat akan mengefisienkan upaya pengamanan. Di samping itu, ada nilai tambah yang sulit dicapai dengan memakai model pengamanan yang tersebar di masing-masing e-Layanan, yaitu pengguna akan dipermudah urusannya. Sekali terdaftar di manajemen pengamanan ini, seorang pengguna dapat memakai e-Layanan lain dari Kemdiknas tanpa perlu mendaftar ulang.

Di sisi Kemdiknas, adanya ID tunggal untuk pengguna ini memungkinkan untuk melakukan rekonsiliasi data dari berbagai e-Layanan, karena dapat dihubungkan melalui ID penggunanya. Hal ini akan menciptakan akuntabilitas untuk sistem-sistem yang selama ini berdiri sendiri-sendiri. Misalnya orang (yang mewakili institusi tertentu) yang terverifikasi untuk perijinan dan akreditasi akan langsung bisa dihubungkan datanya pada saat evaluasi hibah. Seorang pegawai yang akan mendapat

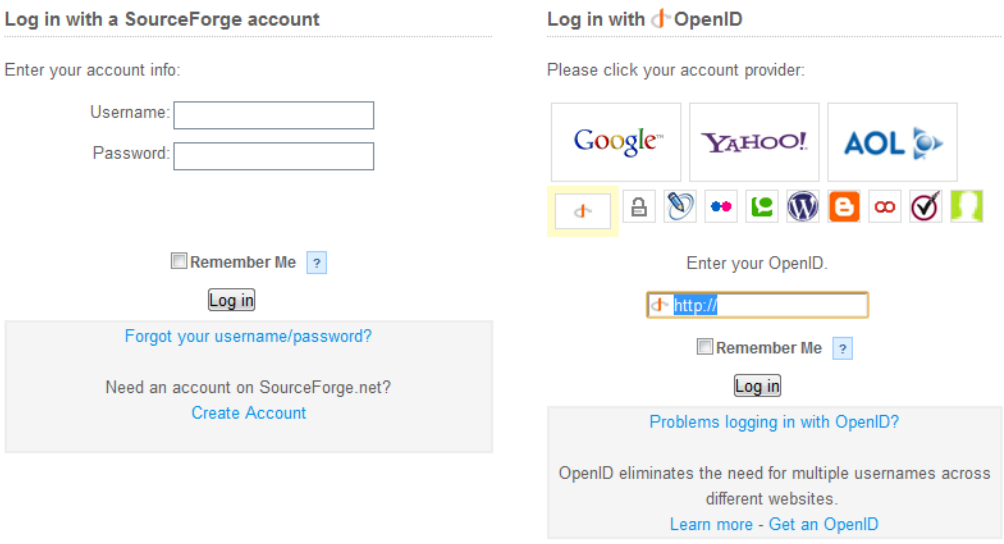
beasiswa ke luar negeri akan bisa di-*track* status berkas tugas belajarnya, dan apabila sudah lulus, penyetaraan ijazahnya dapat dialirkan dan ditindaklanjuti lebih efisien.

4.5.1. Penggunaan *OpenID*[2]

Meskipun sudah ada penyatuan ID, tetap saja "lupa password" merupakan masalah dalam pengelolaan keamanan / akses. Fitur "Reset Password" memang tersedia, tetapi biasanya seorang pengguna akan mendaftarkan diri lagi bila dia juga lupa alamat e-mail yang dipakai untuk mendaftar atau bisa juga *account* e-mailnya telah ditutup. Bagi pengguna publik, *OpenID* menyederhanakan proses pendaftaran dengan memungkinkan pengguna untuk log in *(sign in)* dengan account OpenID yang ada dalam satu klik dan mempercepat proses pendaftaran *(sign up)*.

OpenID adalah fasilitas untuk memanfaatkan account pengguna yang ada di sejumlah penyedia layanan e-mail dan blog terkenal yang menjadi anggota pendukung OpenID, seperti misalnya: Google, Yahoo, Flickr, Blogspot, Myspace dan lain-lain. Dengan memanfaatkan account yang sudah ada tersebut, proses verifikasi pengguna dapat disederhanakan. OpenID meniadakan proses untuk membawa pengguna baru keluar dari situs e-Layanan untuk memverifikasi alamat e-mail.

Di samping itu, bila pengguna sedang membuka layanan Google, Yahoo, Flickr, atau yang lainnya yang accountnya dimanfaatkan untuk mendaftar di e-Layanan, dia dapat langsung masuk ke e-Layanan tanpa harus melewati proses log in lagi. Hal ini dikarenakan dia sudah log in di layanan sebelumnya dan ID-nya sudah diverifikasi.



Gambar 21. Contoh log in layanan Sourceforge.net yang menggunakan OpenID.

[2] http://openid.net

Integrasi OpenID ini ke dalam layanan Manajemen Pengamanan e-Layanan sebagai soft infrastruktur terpusat akan memperkuat dan mempercepat partisipasi masyarakat maupun partisipasi dari unit penyedia e-Layanan. Masyarakat mendapatkan kemudahan dalam pendaftarannya, unit penyedia e-Layanan mendapatkan “captive market” dari pengguna layanan e-mail dan blog yang jumlahnya sangat besar.

BAB V
KESIMPULAN

Dari kajian Manajemen Pengamanan e-Layanan ini, ada beberapa kesimpulan yang dapat diambil yaitu :

1. Fokus reformasi layanan Kemdiknas yang berorientasi pada publik telah membawa kebutuhan lebih banyak dalam hal pengamanan aplikasi, khususnya aplikasi e-Layanan.

2. Kebutuhan untuk bertransaksi dengan publik, unit kerja di daerah serta tuntutan integrasi dengan unit organisasi lain di lingkungan Kemdiknas menyebabkan dibutuhkannya solusi yang merupakan gabungan antara *Role-Based Access Control (RBAC)* dan *Row Level Security (RLS)*.

3. Kebutuhan akan akuntabilitas sistem yang tinggi dapat dipenuhi dengan :

    a. Pembatasan akses per proses/transaksi

    b. Pembatasan akses lingkup data

    c. Pencatatan log operasi per sesi pemakaian.

4. Penggabungan penanganan manajemen keamanan yang terpusat dapat memberikan manfaat lebih. Oleh sebab itu, peluang untuk menjadikan keamanan sebagai *soft infrastructure* dapat dipertimbangkan dengan langkah awal yang berupa penyatuan manajemen akses untuk aplikasi-aplikasi yang baru dibangun.

Daftar Pustaka

1. Ferraiolo, D.F. and Kuhn, D.R. (October 1992). "Role-Based Access Control". 15th National Computer Security Conference. pp. 554–563.

2. Sandhu, R., Coyne, E.J., Feinstein, H.L. and Youman, C.E. (August 1996). "Role-Based Access Control Models". IEEE Computer (IEEE Press) 29 (2): 38–47.

3. Don Thibeau: “Open Trust Frameworks for Open Government: Enabling Citizen Involvement through Open Identity Technologies”, http://openid.net/docs/Open_Trust_Frameworks_for_Govts.pdf, 2009