

## KEAMANAN BATASAN DATA MENGGUNAKAN METODE *WRITE VALIDATION* DALAM DISTRIBUTED DATABASE SYSTEM

Untung Rahardja<sup>1</sup>

Hidayati<sup>2</sup>

Reny Ardyanti<sup>3</sup>

*E-mail: untung.rahardja@faculty.raharja.ac.id, hidayati@faculty.raharja.ac.id,  
renyardyanti@si.raharja.ac.id*

Diterima :22 Juni 2010/Disetujui : 18 Agustus 2010

### **ABSTRACT**

*Along with the advancement in information technology and the rapid rate of development encourages the development of distributed databases. The objective application of web-based information system that is so that users can access information anywhere and anytime. However, increasing numbers of data, making the system take a long time in doing the data display. To handle made of a display buffer which we call by the term Data Mart Queries (DMQ). However, it turns out DMQ any constraints which have the data will not be updated so that the displayed data is not realtime data, while data is always changing all the time in different user and client. For that needed a new solution through the write method of validation. Although the data is not updated display data, but with this method should make the process of validation system first, before his execution on the run. In this article, identified two problems in a distributed system, especially a problem in displaying the data view to users quickly, yet remain secure when executed, write the definition of validation, validation write three characteristics, advantages and disadvantages of write validation, algorithm and benefit from this method. On implementation, the displayed listing program written using ASP script. Contribute to the write method of validation in a distributed database system is a solution which is helpful to keep the time of the execution of database security, although the data shown is the data buffer that is used to maintain the standard TWT (tolerable Wait Time).*

1. **Dosen Jurusan Sistem Informasi, STMIK Raharja**  
Jl. Jend. Sudirman No. 40 Cikokol - Tangerang Telp. 5529692
2. **Mahasiswa Jurusan Sistem Informasi, STMIK Raharja**  
Jl. Jend. Sudirman No. 40 Cikokol - Tangerang Telp. 5529692
3. **Mahasiswa Jurusan Sistem Informasi, STMIK Raharja**  
Jl. Jend. Sudirman No. 40 Cikokol - Tangerang Telp. 5529692

### **ABSTRAKSI**

*Seiring dengan kemajuan teknologi informasi dan laju perkembangannya yang semakin pesat mendorong adanya perkembangan distributed database. Tujuan diterapkannya sistem informasi berbasis web yaitu agar user dapat mengakses informasi dimanapun dan kapanpun. Namun, semakin bertambahnya data, membuat sistem membutuhkan waktu lama dalam melakukan display data. Untuk menanganinya dibuatlah sebuah buffer display yang biasa kita sebut dengan istilah Data Mart Query (DMQ). Namun, ternyata DMQ tersebut pun mempunyai kendala yaitu data menjadi tidak update sehingga data yang ditampilkan bukanlah data realtime, sementara data selalu berubah setiap saat diberbagai user dan client. Untuk itu diperlukan sebuah solusi baru melalui metode write validation. Walaupun display data bukanlah data update, namun dengan metode ini sistem harus melakukan proses validation terlebih dahulu, sebelum eksekusi dijalankan. Dalam artikel ini, diidentifikasi 2 masalah yang dihadapi dalam suatu sistem terdistribusi khususnya masalah dalam menampilkan view data ke pengguna dengan cepat, namun tetap aman saat dieksekusi, definisi dari write validation, 3 ciri khas write validation, keuntungan dan kelemahan dari write validation, algoritma serta manfaat dari metode ini. Pada implementasinya, ditampilkan listing program yang ditulis menggunakan script ASP. Kontribusi metode write validation dalam distributed database system merupakan suatu solusi yang sangat membantu menjaga keamanan database saat dilakukan eksekusi, walaupun data yang ditampilkan merupakan data buffer yang digunakan demi menjaga standard TWT (Tolerable Wait Time).*

*Kata kunci : write validation, DMQ (Data mart Query), TWT (Tolerable Wait Time)*

### **PENDAHULUAN**

Perkembangan teknologi yang terus meningkat dengan pesat, mempengaruhi laju kebutuhan manusia atas informasi, terlebih disuatu organisasi atau perusahaan. Informasi terus mengalir dan jumlahnya semakin lama semakin meningkat seiring dengan jumlah permintaan, serta jumlah data yang semakin banyak, sehingga membuat sistem membutuhkan waktu lama dalam melakukan display data. Selain itu penggunaan database dalam suatu perusahaan dan organisasipun menjadi semakin kompleks terlebih dengan adanya sistem jaringan. Database dapat didistribusikan dari satu komputer ke komputer lain, sehingga jumlah arus pemakaianpun meningkat.

Organisasi maupun perusahaan membutuhkan sistem informasi untuk mengumpulkan, mengolah dan menyimpan data serta mendistribusikan suatu

informasi. Berkembangnya sistem informasi dari waktu ke waktu telah menghasilkan banyak informasi yang semakin kompleks. Kompleksnya informasi tersebut disebabkan oleh banyaknya permintaan, serta meningkatnya jumlah data dan tingkat iterasi perintah display data dalam suatu sistem informasi.

Pemanfaatan teknologi informasi oleh organisasi atau perusahaan secara garis besar bertujuan untuk memudahkan pelaksanaan proses bisnis dan meningkatkan kemampuan kompetitif. Melalui teknologi informasi, diharapkan proses bisnis perusahaan dapat dilaksanakan lebih mudah, cepat, efektif dan efisien. Penggunaan teknologi jaringan didalam suatu organisasi, perguruan tinggi ataupun perusahaan merupakan hal yang biasa di era globalisasi seperti sekarang ini. Didalam suatu sistem jaringan banyak organisasi ataupun perusahaan yang telah menerapkan database terdistribusi, namun semakin bertambahnya data sehingga menimbulkan masalah ketidaknyamanan khususnya perihal respon time agak lama dalam melakukan display data tersebut.

Menurut penelitian yang dilakukan oleh Hoxmeier & DiCesare pada tahun 2000, bahwa waktu respon halaman web adalah metrik paling sederhana yang mengukur kepuasan pengguna. Tapi saat ini *bandwidth* dan pengalaman pengguna telah meningkat, pedoman untuk respon halaman web pun telah berevolusi. Menurut Schneiderman yang didirikan pada tahun 1980, bahwa 15 detik adalah waktu menunggu ditoleransi. Menurut King, bahwa untuk dapat muncul sebagai Internet dan Web di tahun 1990-an, waktu tanggapan toleransi dibagi menjadi 2, yaitu 8 - 12 detik tanpa umpan balik, dan 20 - 30 detik atau lebih dengan umpan balik. Pada tahun 1996 Jacob Nielsen pun menulis bahwa 10 detik adalah waktu respon maksimum sebelum pengguna kehilangan bunga. Zona Riset menerbitkan sebuah studi yang sering dikutip untuk Akamai yang menawarkan aturan 8 detik. Pada awal 2003, King menemukan bahwa rata-rata TWT (*Tolerable Wait Time*) adalah 8,6 detik untuk kondisi sekarang.

Untuk menjaga kepuasan pelanggan dalam mengakses sistem informasi, digunakanlah cara pembuatan sebuah *buffer display* yang biasa disebut dengan *Data Mart Query* (DMQ). DMQ ini akan membantu user dengan cepat melakukan display data, karena data tidak perlu lagi diproses, namun data hanya di *display*. Karena sifatnya yang seperti buffer, maka DMQ ini mempunyai kelemahan yaitu data yang ditampilkan adalah data hasil buffer, bukanlah data *real time*. Sehingga dapat juga dikatakan bahwa data tersebut tidak *update*. Apalagi jika data tersebut adalah data yang selalu berubah setiap saat diberbagai *client*. Untuk itu diperlukan

sebuah metode untuk menangani masalah ini, namun tetap memperhitungkan kepuasan pelanggan dari segi TWT (*Tolerable Wait Time*).

## PERMASALAHAN

*Distributed database* memang memiliki banyak keunggulan terlebih untuk struktur organisasi saat ini. Namun diantara keunggulan itu, *distributed database* juga memungkinkan suatu sistem menjadi lebih kompleks, karena banyaknya database yang tersebar dan terus meningkatnya jumlah data didalam suatu organisasi maupun perusahaan. Jika suatu database memiliki sejumlah data yang tersimpan dengan banyak query dan tabel, suatu permintaan mengakibatkan proses pencarian data atau *source data* menjadi lambat. Selain itu banyaknya *user* dan *client* yang dapat mengakses suatu sistem informasi juga menjadi salah satu faktor penghambatnya. Seperti halnya yang terjadi pada O-OJRS (Otomatisasi Online Jadwal Rencana Study), integritas datanya membuat sistem tersebut melakukan proses lebih lambat.

Didunia perguruan tinggi, KRS adalah salah satu fokus utama agenda setiap semester. Bukan hanya bagi pihak manajemen, namun juga menjadi kebutuhan utama para mahasiswa. Berkembangnya teknologi informasi memaksa kita untuk dapat mengikuti arus tersebut, dan memanfaatkan perkembangannya dalam hal penyusunan Otomatisasi Jadwal Rencana Study (OJRS) serta pendistribusiannya yang dilakukan secara *online*. Tidak berhenti sampai disini, perkembangan teknologi pun menuntut kita agar proses pergantian mata kuliah, yang biasa disebut dengan istilah batal tambah, dapat dilakukan oleh mahasiswa baik secara mandiri maupun *online*.

Untuk menyatakan bahwa batal tambah *online* siap untuk diimplementasikan, itu tidaklah mudah. Banyak hal yang harus ditesting terlebih dahulu. Banyaknya kelas yang dibuka, banyaknya *user* yang mengakses, banyaknya data yang difilter, dan banyaknya *client*, semua itu mempengaruhi proses *testing* tersebut. Hanya sekedar untuk mengakses daftar kelas yang dibuka, setiap *user* membutuhkan waktu kurang lebih 20 detik untuk memprosesnya. Hal ini tidak sesuai dengan teori *response time* yang menyatakan bahwa TWT (*Tolerable Wait Time*) standar adalah 8.6 detik. Untuk itu dinyatakan bahwa lambatnya proses bukan hanya disebabkan banyaknya data, namun juga dikarenakan banyaknya proses filterisasi.

Filterisasi untuk daftar kelas yang dibuka yaitu bahwa data yang di *display* adalah seluruh kelas yang sesuai dengan id kurikulum, yang sesuai dengan shift yang diambil (siang/malam), yang belum lulus, belum ada di KST semester aktif dan KST semester lalu, status jadwalnya tidak bentrok, populasi mahasiswa tidak melebihi quota kelas, serta prasyarat mata kuliah pun sudah memenuhi (sudah lulus).

Untuk mengatasi masalah diatas dan demi mempertahankan kecepatan proses data digunakanlah konsep *Data Mart Query* (DMQ). DMQ (*Data Mart Query*) merupakan konsep yang menerapkan analogi “*Waste Space for Speed*”. Cara kerjanya yaitu melakukan pemisahan antara “*Engine*” dan “*Display*”. Secara umum DMQ menghasilkan sebuah *display* data yang jauh lebih cepat dibandingkan dengan sebelumnya, karena DMQ tidak melakukan proses lagi dalam menampilkan data. Akan tetapi dengan memakai DMQ bukan berarti masalah selesai. Terdapat masalah-masalah lainnya yaitu data menjadi tidak *update* sehingga data yang tampil bukanlah data *realtime*, sementara data selalu berubah setiap saat diberbagai *user* dan *client*. Kronologisnya yaitu ,

NO	KODE KELAS	POPULASI	MATA KULIAH	SKS	HARI	RUANG	MULAI	AKHIR	*
1	SI131C	49/50	Sistem Operasi	3	Senin	M-304	14:40	16:20	<input type="checkbox"/>
2	SI131F	47/50	Sistem Operasi	3	Selasa	M-304	11:20	13:00	<input type="checkbox"/>

Gambar 1. Daftar kelas buka sesuai ID kurikulum pada O-OJRS tambah kelas

Seperti gambar diatas, saat seorang mahasiswa melakukan proses tambah kelas, dinyatakan bahwa salah satu kelas yaitu SI131C memenuhi syarat. Berarti mahasiswa tersebut berhak untuk menambahkan kelas tersebut pada KST nya. Namun apakah populasi mahasiswanya masih memenuhi syarat quota kelas? Karena mahasiswa lain pun memiliki hak yang sama seperti mahasiswa ini untuk menambahkan kelas SI131C itu pada KST nya di *client* lain.

Artinya yaitu antar mahasiswa yang satu dengan mahasiswa yang lainnya saling berlomba untuk dapat terdaftar disetiap kelas. Masalahnya yaitu lebih kepada filterisasi data, bahwa dengan *display* data yang tidak *realtime*, dan dengan ada persaingan proses diberbagai *user* dan *client*, setiap kelas harus tetap dapat terjaga agar batas quotanya tidak terlewati. Dan tetap dapat filterisasi bahwa kelas yang diambil adalah kelas yang tidak bentrok dengan jadwal KST.

Dari pembahasan diatas, dapat dirumuskan 2 permasalahan yaitu sebagai berikut :

1. Metode apakah yang dapat menjaga keamanan batasan data walaupun data *display* tidak *realtime*, sementara data selalu *update* diberbagai *user* dan *client* setiap saat ?

2. Metode apakah yang dapat selalu menjaga standar TWT (*Tolerable Wait Time*) agar tidak melebihi dari 8,6 detik *display data*?

### LITERATURE REVIEW

Banyak penelitian yang sebelumnya dilakukan mengenai *write validation*. Dalam upaya pengembangan *display data* ini perlu dilakukan studi pustaka sebagai salah satu dari penerapan metode penelitian yang akan dilakukan. Diantaranya adalah mengidentifikasi kesenjangan (*identify gaps*), menghindari pembuatan ulang (*reinventing the wheel*), mengidentifikasi metode yang pernah dilakukan, meneruskan penelitian sebelumnya, serta mengetahui orang lain yang spesialisasi dan area penelitiannya sama dibidang ini. Beberapa *Literature review* tersebut adalah sebagai berikut :

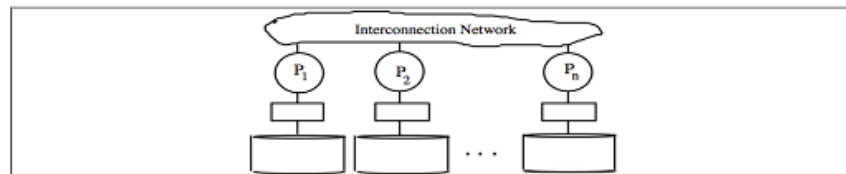
1. Penelitian ini dilakukan oleh Nadezhda Filipova dan Filcho Filipov 2008 dari *University of Economics*. Varna, Bul. Kniaz Boris I berjudul "*Development of database for distributed information measurement and control system*". Penelitian ini menjelaskan mengenai pengembangan *database* dari pengukuran informasi yang didistribusikan dan sistem kontrol yang menerapkan metode optik untuk plasma spectroscopy fisika dan penelitian atom collisions dan menyediakan akses untuk mendapat informasi dan sumber daya perangkat keras di jaringan Intranet/Internet, berdasarkan *database* pada sistem manajemen *database Oracle9i*. Perangkat lunak klien yang diwujudkan adalah dalam *Java Language*. Perangkat lunak ini dikembangkan dengan menggunakan model arsitektur, yang memisahkan aplikasi data dari komponen grafis presentasi dan masukan pengolahan logika. Berikut grafis presentasi telah dilaksanakan, pengukuran radiasi dari Spectra beam plasma dan benda, perangsangan fungsi *non-elastic collisions* dari berat partikel dan analisis data yang diperoleh dalam percobaan sebelumnya. Berikut grafis klien yang memiliki fungsi interaksi dengan *database browsing* informasi tentang percobaan dari jenis tertentu, pencarian data dengan berbagai kriteria, dan memasukkan informasi tentang validasi percobaan sebelumnya[1].
2. Penelitian yang dilakukan oleh Lieven Desmet, Bart Jacobs, Frank Piessens, Wouter Joosen, 2004, dari Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS), tentang *A Generic Architecture for Web Applications to Support Threat Analysis of Infrastructural Components*. Penelitian ini menjelaskan bahwa tentang analisis ancaman yang berguna dari platform aplikasi web, beberapa arsitektur asumsi tentang aplikasi tersebut harus dibuat. Dokumen ini menjelaskan generik khas

arsitektur aplikasi web. Ini berfungsi sebagai dasar untuk menganalisis ancaman pada infrastruktur yang paling penting komponen dalam arsitektur[2].

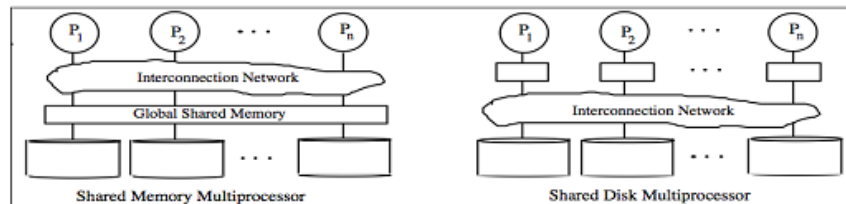
3. Penelitian ini dilakukan oleh Jun Lin Lin dan Margaret H. Dunham dari Southern Methodist University dan Mario A. Nascimento berjudul “*A Survey of Distributed Database Checkpointing*”. Penelitian ini membahas mengenai *checkpointing* pada *database* terdistribusi dan pendekatan-pendekatan yang digunakan. Penelitian ini bermula dari adanya banyak survey yang dilakukan berkenaan dengan proses *recovery database*, dan banyak teknik yang diusulkan untuk mengatasinya. Dengan *distributed database checkpointing*, dapat mengurangi waktu proses *recovery* suatu kegagalan didalam *database* terdistribusi. *Checkpointing* dapat digambarkan sebagai suatu aktivitas menulis informasi ke penyimpanan yang stabil selama operasi normal dalam rangka mengurangi jumlah pekerjaan pada saat *restart*. Penelitian ini membantah bahwa sedikit batasan dan sedikit sumber daya menjadi masalah dalam pendekatan *database* terdistribusi, serta Membantah bahwa *checkpointing* hanya dapat digunakan untuk sistem distribusi yang *multidatabase*. Meskipun penelitian ini telah banyak dilakukan namun cukup rumit dalam implementasinya. Dengan penelitian ini kita dapat mengembangkan *database* terdistribusi dengan *checkpointing* untuk mempercepat proses *recovery database*[3].

Penelitian ini dilakukan oleh David J. DeWitt dari Universitas Wisconsin dan Jim Gray tahun 1992 berjudul “*Parallel Database Systems: The Future of High Performance Database Processing*”. Penelitian ini dilakukan dengan konsep *database* terdistribusi yang merupakan *database* yang disimpan pada beberapa komputer yang terdistribusi satu sama lain. Pada penelitian ini, dijelaskan Sistem *database* paralel mulai menggantikan *Mainframe* komputer besar untuk pengolahan data dan transaksi tugas. Paralel *database* komputer memiliki arsitektur yang berkembang dari penggunaan perangkat lunak yang eksotik untuk perangkat keras yang paralel. Seperti kebanyakan aplikasi, *user* menginginkan *hardware* sistem *database* yang murah, cepat. Ini menyangkut tentang prosesor, memori dan disk. Akibatnya, konsep *hardware database* yang eksotis tidak sesuai untuk teknologi saat ini. Di lain sisi, ketersediaan *microprocessors* cepat, murah dan kecil menjadi paket standar murah tapi cepat sehingga menjadi *platform* yang ideal untuk sistem *database* paralel. Stonebraker mengusulkan rancangan sederhana untuk spektrum disain yaitu *shared memory*, *shared disk* dan *shared nothing*. Dan bahasa yang digunakan dalam *database* adalah SQL sesuai dengan standar ANSI dan ISO.

Dengan penelitian ini, kita dapat mengembangkan sistem *database* agar dapat digunakan diberbagai ruang lingkup[4].



**Figure 3. The basic shared-nothing design.** Each processor has a private memory and one or more disks. Processors communicate via a high-speed interconnect network. Teradata, Tandem, nCUBE, and the newer VAXclusters typify this design.



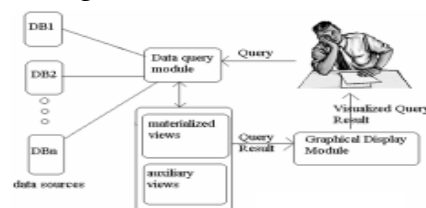
**Figure 4. The shared-memory and shared-disk designs.** A shared-memory multi-processor connects all processors to a globally shared memory. Multi-processor IBM/370, VAX, and Sequent computers are typical examples of shared-memory designs. Shared-disk systems give each processor a private memory, but all the processors can directly address all the disks. Digital's VAXcluster and IBM's Sysplex typify this design.

Gambar 2. Desain Shared-Nothing, Shared-Memory dan Shared-Disk

1. Penelitian yang dilakukan oleh Steven P. Coy dari *University of Maryland* berjudul "*Security Implication of the Choice of Distributed Database Management System Model: Relational Vs Object Oriented*". Penelitian ini menjelaskan bahwa keamanan data harus dibenahi ketika mengembangkan *database* dan diantaranya memilih antara *relational* dan *object oriented model*. Banyak faktor yang harus dipertimbangkan, terutama dari segi efektifitas dan efisiensi, juga apakah sekuritas dan integritas ini memakan sumber daya yang terlalu besar tidak semata mata fitur keamanan. Kedua pilihan ini akan mempengaruhi kekuatan dan kelemahan dari *database* tersebut. Untuk *centralized database* kedua model ini bisa dikatakan sama baiknya. Namun untuk *distributed database*, *relational model* lebih unggul dibidang sekuritas. Ini lebih banyak disebabkan karena *object oriented model database* masih kurang maturitasnya. Sehingga didalam lingkungan heterogenous, proses integritasnya masih menimbulkan banyak masalah. OODBMS tetap saja masih perlu perkembangan teknologi lebih lanjut, namun di lingkungan homogenous, OODBMS dapat menjadi pilihan yang baik[5].

2. Penelitian yang dilakukan oleh Stephane Gançarski, Claudia León, Hubert Naacke, Marta Rukoz and Pablo Santini yang berjudul “*Integrity Constraint Checking in Distributed Nested Transactions over a Database Cluster*” adalah sebuah solusi untuk memeriksa integritas dan kendala global dalam berhubungan multi *database* sistem. Penelitian ini juga menyajikan hasil eksperimental yang diperoleh atas solusi *PC cluster* dengan Oracle9i DBMS. Tujuan adalah melakukan eksperimentasi untuk mengukur waktu yang dihabiskan dalam memeriksa kendala global dalam sistem yang terdistribusi. Alhasil menunjukkan bahwa *overhead* berkurang hingga 50% dibandingkan dengan pemeriksaan integritas yang terpusat. Studi menunjukkan bahwa sistem berkemungkinan besar melanggar *referential integrity* dan *global conjunctive constraints*. Namun dengan cara *distributed nested transactions*, dengan adanya eksekusi dan parallelism, integritas dapat lebih terjamin[6].

Penelitian yang dilakukan oleh Lubomir Stanchev dari University of Waterloo tahun 2001 berjudul “*Semantic Data Control In Distributed Database Environment*”. Penelitian ini menyatakan bahwa ada tiga tujuan utama dalam *semantic data control* yaitu: *view managemen*, *data security* dan *semantic integrity control*. Dalam sebuah relasi, fungsi-fungsi ini dapat mencapai keseragaman dengan menegakkan aturan-aturan manipulasi kontrol data. Solusinya adalah dengan sentralisasi ataupun terdistribusi. Dua hal utama yang efisien untuk melakukan kontrol adalah definisi data dan penyimpanan aturan (situs pilihan) dan penegakan desain algoritma yang meminimalkan biaya komunikasi. Masalahnya adalah sulit, karena peningkatan fungsi (dan umum) cenderung meningkatkan komunikasi situs. Solusi untuk semantik data kontrol terdistribusi adalah eksistensi dari sentralisasi solusi. Masalahnya adalah sederhana jika aturan kontrol sepenuhnya direplikasi di semua situs dan sulit jika situs otonomi dipatenkan. Selain itu, khusus optimasi dapat dilakukan untuk meminimalkan biaya kontrol data tetapi dengan tambahan *overhead* seperti pengelolaan data *snapshot*. Dengan demikian, spesifikasi kontrol data terdistribusi harus disertakan pada desain *database* sehingga biaya kontrol update untuk program-program ini juga dipertimbangkan[7].



Gambar 3. Data Visualization dengan materialized and auxiliary views

1. Penelitian ini dilakukan oleh Allison L. Powell James C.dkk, Perancis Departemen Ilmu Komputer Universitas Virginia, berjudul berjudul “*The Impact of Database Selection on Distributed Searching*”. Penelitian ini menjelaskan bahwa *distributed searching* terdiri dari 3 bagian yaitu *database selection*, *query processing*, dan *results merging*. Cukup beberapa *database* yang dijadikan *database* seleksi (tidak semuanya) dan performa akan meningkat cukup signifikan. Bila seleksi *database* dilakukan dengan baik, pencarian secara *distributed* akan berkinerja lebih baik dibandingkan pencarian secara sentralisasi. Pencarian *database* juga ditambahkan proses seleksi dan ranking sehingga secara potensial meningkatkan efektifitas pencarian data[8].
2. Penelitian ini dilakukan oleh Shadi Aljawarneh, Faisal Alkhateeb and Eslam Al Maghayreh yang berjudul “*A Semantic Data Validation Service for Web Applications*” dari Universitas Yarmouk. Penelitian ini menjelaskan bahwa Sebuah validasi Input bisa menjadi masalah kritis. menyebabkan kegagalan dalam perangkat lunak dan juga dapat mematahkan keamanan pada aplikasi web seperti akses tidak sah ke data. Sekarang, diperkirakan aplikasi web kerentanan (seperti XSS atau SQL injection) selama lebih dari dua pertiga dari keamanan web melaporkan kerentanan. Solusi untuk mencegahnya validasi baru layanan data yang didasarkan pada Web semantik Technologies, telah dirancang dan dilaksanakan untuk mencegah kerentanan keamanan web di aplikasi tingkat dan untuk mengamankan sistem web bahkan jika modul input validasi yang dilewati akan melewati beberapa eksekusi. Dengan demikian Dalam keamanan aplikasi web, validasi data adalah proses untuk memastikan bahwa aplikasi web beroperasi pada bersih, data yang benar dan berguna .Menggunakan aturan validasi yang memeriksa kebenaran, kebermaknaan, dan keamanan data yang masukan ke sistem berbasis web [9].
3. Penelitian yang dilakukan oleh Hamidah Ibrahim, “*Deriving Global Integritas Dan Local Rules For Distributed Database*”. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Putra Malaysia, 43400 UPM Serdang. Ibrahim mengatakan bahwa tujuan terpenting didalam *database* sistem adalah menjamin konsistensi data, yang berarti bahwa data yang terdapat dalam *database* harus baik dan akurat. Didalam pelaksanaannya untuk menjaga konsistensi perubahan data sangat sulit, khususnya untuk didistribusikan dalam *database*. Dalam tulisan ini, menjelaskan sebuah algoritma penegakan aturan berdasarkan mekanisme

untuk didistribusikan *database* yang bertujuan meminimalisir jumlah data yang harus ditransfer atau diakses diseluruh jaringan yang menjaga konsistensi dari *database* di satu situs, yaitu di situs mana pembaruan perlu dilakukan. Teknik ini disebut sebagai tes integritas generasi, yang berasal dari lokal dan global integritas, dan aturan yang telah efektif dapat mengurangi biaya kendala dalam memeriksa suatu data yang telah didistribusikan dalam lingkungan. Didalam penelitian ini telah berhasil menghasilkan sebuah sistem sentralistik yang besar dengan tingkat kehandalan yang tinggi untuk integritas data [10].

4. Penelitian ini dilakukan oleh Untung Rahardja dan Shakinah Badar dari Perguruan Tinggi Raharja Indonesia yang berjudul "*Penerapan metode Data Mart Query (DMQ) dalam Distributed Database System*" tahun 2009. Penelitian ini menjelaskan *Distributed database* memiliki banyak keunggulan terlebih untuk struktur organisasi saat ini. Namun, karena banyaknya database yang tersebar dan jumlah data yang banyak dan terus meningkat didalam suatu organisasi maupun perusahaan. Jika suatu database memiliki sejumlah data yang tersimpan dengan banyak query dan tabel, suatu permintaan mengakibatkan proses pencarian data atau source data menjadi lambat. Selain itu banyaknya user yang dapat mengakses suatu tampilan web atau Web display suatu sistem informasi juga menjadi lambat. Dengan adanya *Data Mart Query (DMQ)* merupakan metode yang tepat untuk mempercepat waktu proses pada suatu sistem informasi dengan database yang terdistribusi. DMQ ini digunakan untuk menghindari penggunaan Query majemuk. Dengan demikian DMQ akan mengorbankan besarnya kapasitas penyimpanan data (*space hard disk*) untuk meningkatkan kecepatan (*increase speed*) dalam pengaksesan. Hal ini pun telah dibuktikan baik secara logik, secara grafik dengan perhitungan regresi linear dan korelasi linear dan juga melalui implementasi [11].

Dari sebelas *literature review* yang ada, telah banyak penelitian mengenai *checkpointing*, *parallel database system*, pembahasan *component database system*, juga mengenai *security*. Disamping itu juga ada pembahasan mengenai *nested transaction*, *distributed searching*, *view management*, *fragment allocation*, dan juga *DMQ*. Namun dapat disimpulkan pula bahwa belum ada peneliti yang secara khusus membahas atau mengatasi masalah perihal batasan keamanan data melalui proses *write validation*.

## PEMECAHAN MASALAH

Untuk mengatasi permasalahan seperti yang telah dijelaskan diatas, dapat dilakukan melalui penerapan metode *Write Validation*. Berikut merupakan 3 ciri khas dari *Write Validation* yang diterapkan pada proses tambah kelas Otomatisasi Online Jadwal Rencana Study (O-OJRS) :

1. Adanya proses validasi sebelum eksekusi dijalankan.
2. Jika salah satu syarat tidak terpenuhi, maka proses eksekusi dibatalkan.
3. Jika seluruh syarat terpenuhi, maka proses tambah kelas dapat dieksekusi.

Untuk menangani masalah keamanan batasan data pada proses tambah kelas O-OJRS, diperlukan suatu proses validasi guna melakukan pengecekan terhadap seluruh status batasan data. Batasan-batasan yang digunakan pada proses tambah kelas O-OJRS yaitu sebagai berikut :

1. Sesuai dengan kurikulum.
2. Belum tercatat didaftar nilai sebagai matakuliah yang lulus, dan tidak sedang diambil pada KST semester aktif.
3. Mata kuliah prasyarat sudah lulus atau sedang diambil pada KST semester aktif.
4. Populasi mahasiswa belum melebihi batas quota kelas.
5. Tidak bentrok dengan KST.

Untuk batasan 1, 2, dan 3, merupakan batasan yang sifatnya relatif tetap. Biasanya berubah minimal semester sekali. Sehingga tidak terlalu berpengaruh walaupun pengecekan telah menggunakan metode *Write Validation*.

Lain halnya dengan metode 4 dan 5, yang sifatnya terus menerus berubah diberbagai *user* dan *client*. Saat seorang mahasiswa A menyatakan OK akan menambahkan sebuah kelas pada KST nya, maka syarat 4 dan 5 ini yang penting untuk divalidasi. Benarkah kelas tersebut tidak bentrok dengan KSTnya? Dan benarkah populasi mahasiswa belum melebihi batas quota kelas? Apabila populasi mahasiswa tercatat 50/51. Itu memiliki arti bahwa populasinya telah terisi 50 mahasiswa, dari batas quota yaitu 51. Berarti hanya tersisa 1 mahasiswa lagi untuk mengisi kelas tersebut. Mahasiswa lain yaitu mahasiswa B melalui *client* lain dapat mengisi lebih dahulu populasi kelas tersebut, dan menggenapkan quotanya menjadi 51/51. Sehingga mahasiswa A dinyatakan ditolak, tidak dapat menambahkan kelas tersebut dikarenakan status quota penuh. Hal ini sesuai dengan point nomor [1] dan [2].

Lain halnya jika segala prasyarat terpenuhi, maka proses penambahan kelas baru dapat dieksekusi. Hal ini sesuai dengan point nomor [3].

## IMPLEMENTASI

Penanganan masalah keamanan batasan data menggunakan metode *Write Validation* sudah diimplementasikan pada Perguruan Tinggi Raharja, yakni pada sistem informasi SIS O-OJRS (Otomatisasi Online Jadwal Rencana Studi). *Students Information Services* atau yang biasa disingkat SIS, merupakan sistem yang dikembangkan oleh Perguruan Tinggi Raharja dengan tujuan sebagai sistem pelayanan informasi kepada mahasiswa secara optimal. Pengembangan SIS juga merupakan akses publikasi bagi Perguruan Tinggi Raharja di bidang ilmu komputer dan dunia IT khususnya.

SIS sudah dikembangkan ke dalam beberapa versi, dimana masing-masing merupakan kelanjutan dari SIS versi sebelumnya. SIS O-OJRS (Otomatisasi Online Jadwal Rencana Studi) merupakan SIS versi ke-4.2. Sesuai namanya, SIS O-OJRS dibuat untuk kebutuhan batal tambah KST mahasiswa secara online.

Hasil akhir yang ingin dicapai dari SIS O-OJRS ini yaitu dihasilkanlah sebuah sistem yang secara flexibel melayani segala proses batal tambah KST mahasiswa secara online dan dapat dilakukan secara mandiri. Prosesnya terdiri dari tambah kelas, pindah kelas, dan hapus kelas. Karena itulah, untuk dapat meningkatkan pelayanan terhadap mahasiswa, diterapkanlah metode *Write Validation* ini.



Gambar 4. Biodata Mahasiswa pada SIS

Gambar 4 diatas merupakan tampilan biodata mahasiswa. Dari halaman tersebut dapat diketahui segala informasi mengenai seorang mahasiwa, baik mengenai data pribadinya maupun mengenai data akademiknya. Untuk melakukan batal tambah KST, silahkan masuk pada menu O-OJRS. Akan ditampilkan tombol-tombol berbeda sesuai dengan status mahasiswa tersebut. Jika seorang mahasiswa dinyatakan belum membayar registrasi perkuliahan, maka pada menu O-OJRS hanya akan ditampilkan 1 tombol yaitu tombol JRS AKTIF. Namun jika seorang mahasiswa dinyatakan sudah registrasi, maka akan ada 3 tombol berbeda yang diklasifikasikan kembali berdasarkan status KST nya. Jika KST nya dinyatakan belum final, akan

ditampilkan tombol JRS Aktif, tombol KST Aktif, dan tombol O-OJRS. Sementara jika mahasiswa tersebut dinyatakan sudah melakukan proses finalisasi pada KST, maka akan ditampilkan 3 tombol yang berbeda pula meliputi tombol JRS Aktif, tombol KSTF, dan tombol RBK. Untuk melakukan batal tambah *online*, dapat dilakukan dengan mengklik tombol O-OJRS. Berikut adalah tampilannya:

BACK

**KARTU STUDI TETAP (KST)**

Nomor Induk Mahasiswa : 0814462048	Jurusan : Sistem Informasi
Nama Mahasiswa : Jetty Susanti	Konsentrasi : Komputer Akuntansi
Jenjang Studi : Sarjana	Shift : Siang

NO	KELAS	MATA KULIAH	SKS	SHIFT	HARI	RUANG	WAKTU	KET		
1	TI999A	Independent Study	2	-	-	-	00:00-00:00	OK		
2	JR111D	Jaringan Komputer	3	Siang	Rabu	M-201	09:40-11:20	OK		
3	JR111DP	Jaringan Komputer (Praktek)	0	Siang	Rabu	M-201	11:20-12:10	OK	-	-
4	UM140B	Kewarganegaraan	2	Siang	Kamis	M-206	10:30-12:10	OK		
5	BI103D	Bahasa Indonesia	2	Siang	Kamis	M-104	12:10-13:50	OK		
6	MJ110D	Manajemen Proyek	2	Siang	Jumat	M-201	09:40-11:20	OK		
7	AK111AP	Akuntansi Biaya (Praktek)	0	Siang	Jumat	M-103	13:00-13:50	OK	-	-
8	AK111A	Akuntansi Biaya	3	Siang	Jumat	M-103	13:50-15:30	OK		

Jumlah Satuan Kredit Semester (SKS) = 14 sks

**Keterangan :**  
**OK :** Jadwal tidak bermasalah.  
**Bentrok :** Ada jadwal yang dijalankan dalam waktu bersamaan.  
**Confirm :** Khusus untuk KKP, TA/Skripsi yang statusnya masih confirm, berarti masih ada prasyarat yang belum lengkap. Harus konfirmasi kepada RPU.

FINAL

Gambar 5. O-OJRS batal tambah

Metode *write validation* diterapkan pada proses tambah kelas O-OJRS. Untuk masuk kedalam fasilitas tambah kelas, silahkan klik di tombol plus (+) seperti pada gambar 5 diatas. Akan ditampilkan kelas-kelas yang telah diproses sebelumnya. Kelas-kelas tersebut telah memenuhi persyaratan sebagai berikut :

1. Sesuai dengan kurikulum.
2. Belum tercantum didaftar nilai sebagai matakuliah yang lulus, dan tidak sedang diambil pada KST semester aktif.
3. Mata kuliah prasyarat sudah lulus atau sedang diambil pada KST semester aktif.
4. Populasi mahasiswa belum melebihi batas quota kelas.
5. Tidak bentrok dengan KST.

Berikut adalah daftar kelasnya :

BACK

Daftar kelas buka sesuai ID Kurikulum: BACK KST

NO	KODE KELAS	POPULASI	MATA KULIAH	SKS	HARI	RUANG	MULAI	AKHIR	*
1	SI131C	49/50	Sistem Operasi	3	Senin	M-304	14:40	16:20	<input type="checkbox"/>
2	SI131F	47/50	Sistem Operasi	3	Selasa	M-304	11:20	13:00	<input type="checkbox"/>
3	SI138B	38/39	Sistem Basis Data	3	Rabu	M-201	13:50	15:30	<input type="checkbox"/>
4	SI138C	38/39	Sistem Basis Data	3	Rabu	M-206	15:30	17:10	<input type="checkbox"/>
5	SI138BP	38/39	Sistem Basis Data (Praktek)	0	Rabu	M-102	15:30	16:20	
6	SI138CP	38/39	Sistem Basis Data (Praktek)	0	Rabu	M-106	14:40	15:30	
7	EK131A	43/50	Matematika Ekonomi	3	Selasa	M-201	09:40	11:20	<input type="checkbox"/>
8	EK131AP	43/50	Matematika Ekonomi (Praktek)	0	Selasa	M-201	11:20	12:10	
9	MJ183B	49/50	Rekayasa Piranti Lunak	3	Senin	M-206	09:40	11:20	<input type="checkbox"/>
10	MJ202B	50/51	Komputer dan Masyarakat	2	Jumat	M-206	08:00	09:40	<input type="checkbox"/>
11	SI121B	49/50	Sistem Informasi Manajemen	3	Senin	M-304	09:40	11:20	<input type="checkbox"/>
12	SI360A	14/50	Sistem Penunjang Keputusan	2	Senin	M-207	11:20	13:00	<input type="checkbox"/>
13	MJ200B	50/52	Etika Profesi	2	Selasa	M-206	13:50	15:30	<input type="checkbox"/>
14	MJ201F	28/50	Kecakapan Antar Personal	2	Sabtu	M-206	12:10	13:50	<input type="checkbox"/>

Gambar 6. Daftar kelas buka sesuai ID kurikulum pada O-OJRS tambah kelas

Pada gambar 6 diatas dijelaskan bahwa salah satu kelas yaitu SI131C memenuhi 5 persyaratan. Dinyatakan bahwa populasinya 49/50, artinya yaitu telah terisi 49 mahasiswa dari quota 50 mahasiswa. Yang berarti hanya dapat diisi oleh

satu mahasiswa lagi. Jika ternyata mahasiswa atas nama Jetty ini berpikir terlalu lama, dan bingung ingin menambahkan kelas yang mana. Sementara ada mahasiswa lain dengan menggunakan client lain melakukan tambah kelas pada kelas SI131C, sehingga genap populasinya memenuhi quota menjadi 50/50. Maka saat Jetty melakukan tambah kelas pada kelas SI131C, sistem akan melakukan pengecekan untuk 5 kriteria diatas.

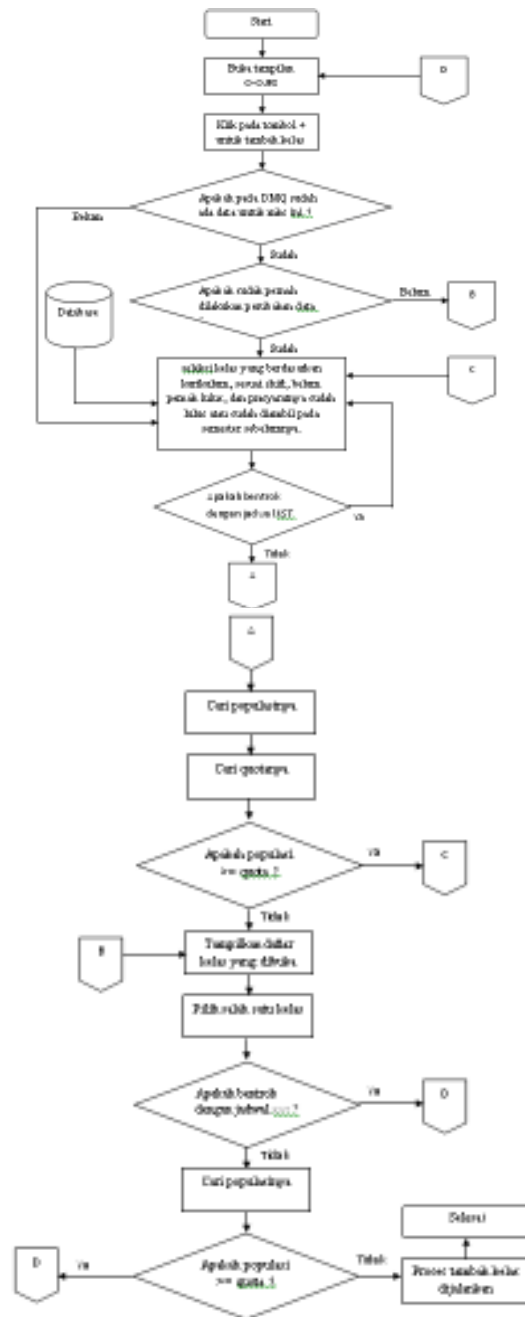
Syarat 1 dan 3 sudah pasti LULUS, karena data kurikulum 90% merupakan data statik yang hanya berubah pada kondisi tertentu. Syarat 2 pun sudah pasti LULUS, karena sifatnya yang semi statik. Perubahan datanya hanya terjadi satu semester sekali yaitu pada saat dikeluarkannya KHS oleh pihak manajemen. Sementara untuk syarat ke-4, yaitu mengenai quota. Syarat ini yang sifatnya sangat dinamik, dan terus berubah dalam hitungan detik. Setiap *user* mempunyai hak untuk dapat melakukan perubahan nilai populasi kelas melalui berbagai *client*, seperti gambaran kasus batal tambah diatas. Untuk itu, pada kasus batal tambah jetty diatas dinyatakan ditolak, dikarenakan populasi kelasnya sudah sama dengan nilai quota kelasnya, dan proses akan dikembalikan pada tampilan gambar 5.

Syarat ke-5 yaitu mengenai bentrok. Sama halnya dengan syarat ke-4, syarat ke-5 ini sifatnya dinamik. Untuk itu perlu dilakukan validasi bahwa kelas yang dipilih statusnya tidak bentrok dengan jadwal KST yang ada. Jika ditemukan jadwal bentrok, maka proses akan di stop, dan akan dikembalikan pada tampilan gambar 5. Namun jika 0% bentrok, maka proses akan dieksekusi pada tahap selanjutnya.

a. Algoritma

```
Var  
Char nim,kode_mk;  
Int kelas,shift;  
Main()  
{  
*/ untuk cek bentrok
```

```
rs = seleksi * from tabel daftar kelas where nim=nim and kode_mk like
kode_mk% and kelas=kelas
while not rs.eof
    rs2 = seleksi * from tabel KST where nim=nim and waktunya = waktu
pada rs
    jika ditemukan jadwal bentrok
        Back to KST.asp?nim=nim
    end if
wend
rs3 = seleksi * from tabel daftar kelas where kode_mk like kode_mk% and
kelas=kelas and jam=1
while not rs3.eof
    jika right(kode_mk,1)="P" then
        kk2=left(kode_mk,5)& (kelas)&"P"
    else
        kk2= left(kode_mk,5)& (kelas)&"P"
    end if
    */ untuk populasi dan quota kelas
    jika right(kode_mk,1)<>"P" and kode_mk<>"TI999" and
kode_mk<>"TA100" and kode_mk<>"TA101" and
kode_mk<>"KP100" then
        rs4 = seleksi * from tabel populasi where kode_kelas=kk2
        rs5 = seleksi * from tabel Quota where kode_kelas=kk
        If populasi pada rs4 >= Quota pada rs5 then
            Back to KST.asp?nim=nim
        end if
    end if
EXECUSI DIJALANKAN
Wend
```



Gambar 7. Flowchart

## c. Source Code

1. *Source code* pada saat klik tombol + untuk melakukan tambah kelas

```

<!--#include File="database.asp"-->
<%
server.ScriptTimeout=4000
nim=trim(request("nim"))
shft=trim(request("shft"))
'untuk cek apa pada DMQ sudah ada data untuk mahasiswa yang
bersangkutan
Sql_Cek2="select * from bt_tbh_kls where nim=" &nim&""
set rs_Cek2=conn.execute(Sql_Cek2)
if not rs_Cek2.eof then
    'bt_log_jml_proses untuk cek apakah telah dilakukan perubahan
data
    Sql_proses5="select * from sysobjects where xtype='U' and
Name='bt_log_jml_proses'"
    set rs_proses5=conn.execute(Sql_proses5)
    if not rs_proses5.eof then
        Sql_proses6="select Jml from bt_log_jml_proses where
nim=" &nim&""
        set rs_proses6=conn.execute(Sql_proses6)
        if rs_proses6.eof then

Response.Redirect("daftar_kelas_tambah_id.asp?nim=" &nim)
        else
            if rs_proses6("Jml")=0 then

Response.Redirect("daftar_kelas_tambah_id.asp?nim=" &nim)
            else
                Sql_delete="delete from bt_tbh_kls where
nim=" &nim&""
                set rs_delete=conn.execute(Sql_delete)
            end if
        end if
    else
        Response.Redirect("daftar_kelas_tambah_id.asp?nim=" &nim)

```

```

end if
end if

```

*'untuk seleksi kelas yang berdasarkan kurikulum, sesuai shift, belum pernah lulus, dan prasyaratnya sudah lulus atau sudah diambil pada semester sebelumnya*

```

sqla="select distinct kode_mk from BT_12_8_SM where
shift="&shft&" and kode_mk not in (select kode_mk from
View_BT_KST2 where nim="&nim&") and kode_mk in (select
kode_mk from view_mkul_id_kurikulum where nim="&nim&") and
kode_mk not in (select kode_mk from dmq_daftar_nilai where
nim="&nim&" and nilai >= 1.70) and kode_mk not in (select kode_mk
from view_mkul_id_kurikulum where nim="&nim&" and Syarat1 < '-'
and (Syarat1 not in (select kode_mk from dmq_daftar_nilai where
nim="&nim&" and nilai >= 1.70) or Syarat1 not in (select kode_mk
from qsub_krs where nim="&nim&" and ta in (select ta from ao_aktif)
and semester in (select semester from ao_aktif))) and (Syarat2 not in
(select kode_mk from dmq_daftar_nilai where nim="&nim&" and nilai
>= 1.70) or Syarat1 not in (select kode_mk from qsub_krs where
nim="&nim&" and ta in (select ta from ao_aktif) and semester in (select
semester from ao_aktif)))) and jam=1 order by kode_mk"
set rsa=conn.execute(sqla)

```

```

while not rsa.eof

```

*'untuk cek bentrok*

```

sqlb="select distinct kode_mk,kelas from BT_12_8_SM where
shift="&shft&" and kode_mk="&trim(rsa("kode_mk"))&" and
jam=1 and right(No_Unit,3) not in (select right(No_Unit,3) from
View_BT_KST2 where NIM="&nim&" and No_Unit is not null)
order by kelas"

```

```

set rsb=conn.execute(sqlb)

```

```

while not rsb.eof

```

```

sqlc="select kode_mk,kelas,no_unit from BT_12_8_SM where
kode_mk="&trim(rsb("kode_mk"))&"P' and
kelas="&rsb("kelas")&" and jam=1 and sks=0"

```

```

set rsc=conn.execute(sqlc)

```

```

if not rsc.eof then

```

```

sqld="select right(No_Unit,3) from View_BT_KST2 where
NIM=""&nim&" and
right(No_Unit,3)=""&right(trim(rsc("No_Unit")),3)&"
set rsd=conn.execute(sqld)
if rsd.eof then
    sql="select * from BT_12_8_SM where kode_mk like
        ""&trim(rs("kode_mk"))&"%' and
        kelas=""&rs("kelas")&" and jam=1 order by sks desc"
    set rs=conn.execute(sql)
    while not rs.eof
        if rs("sks")=0 then
            kk=left(trim(rs("kode_mk")),5)&kel(rs("kelas"))&"P"
            else
                kk=trim(rs("kode_mk"))&kel(rs("kelas"))
            end if
            sql_kurikulum="select semester from
                master_data_kurikulum where
                id_kurikulum=""&trim(rs_id_kurikulum("id_kurikulum"))&"
            and
            kode_mk=""&left(trim(rs("kode_mk")),5)&"
            set rs_kurikulum=conn.execute(sql_kurikulum)

            sql2="select * from BT_12_8_SM where
                kode_mk=""&rs("kode_mk")&" and
                kelas=""&rs("kelas")&" and jam=2"
            set rs2=conn.execute(sql2)
            'untuk cek populasi mahasiswa
            sql3="select * from BT_14_1 where
kode_kelas=""&kk&"
            set rs3=conn.execute(sql3)
            JumKelas=1
            Sql4="select * from View_BT_KST2 where
                NIM=""&nim&" and Hari=""&rs("Hari")&"
            and
            Mulai=""&cdate(hour(rs("Mulai"))&':'&minute(rs("Mulai"))
            &':'&second(rs("Mulai"))&'" order by
                Kode_MK,Jam"

```

```

set rs4=conn.execute(Sql4)
While not rs4.eof
  If not rs4.eof then
    JumKelas=JumKelas+1
  end if
rs4.movenext
wend
'untuk cek quota kelas
sql5="select * from CT_Quota where
kode_kelas=""&kk&""
set rs5=conn.execute(sql5)
if rs("hari")="" then
  Color="#FFFFFF"
  Maks=""~"
  Mx=0
elseif rs("hari")<>"" then
  if rs3("jum_mhs")>=rs5("maksimal") then
    Color="#95CAFF"
    Maks=rs5("maksimal")
    Mx=Maks
  elseif rs3("jum_mhs")<rs5("maksimal") then
    Color="#FFFFFF"
    Maks=rs5("maksimal")
    Mx=Maks
  end if
end if
'cek kondisi antara populasi kelas dengan
quota
if rs3("jum_mhs") < Mx then
  Sql_MK="select Mata_Kuliah from
  Mata_Kuliah where
  Kode=""&trim(rs("kode_mk"))&""
  set rs_MK=conn.execute(Sql_MK)
  if rs("hari")="" then
    Color2="#FF8A8A"
    T="Confirm"
  elseif rs("hari")<>"" then
    If JumKelas>1 then

```



```

,'"&trim(rs("hari"))&"',,'"&trim(rs("ruang"))&"',,'"&rs("
Mulai")&"',,'"&rs2("akhir")&"',,'"&rs_kurikulum("semes
ter")&"',,'"&trim(Gambar)&"',,'"&trim(Color)&"')"
response.write(Sql_insert&"<br>")
set rs_insert=conn.execute(Sql_insert)
end if
rs.movenext
wend
end if
end if
rsb.movenext
wend
rsa.movenext
wend
%>

```

2. *Source code* tampilan daftar kelas yang dibuka

```

<!--#include File="database.asp"-->
<%
server.ScriptTimeout=4000
nim=trim(request("nim"))
'cek jumlah sks
jum_sk=0
Sql_sks="select distinct * from View_BT_KST2 where nim="&nim&"
and jam=1 and sks <> 0"
set rs_sks=conn.execute(Sql_sks)
while not rs_sks.eof
jum_sk=int(jum_sk)+int(rs_sks("sks"))
rs_sks.movenext
wend

if jum_sk=18 then
Sql="select TOP 20 * from bt_tbh_kls where nim="&nim&" and
sks=2 order by semester,kode_mk,kelas,sks desc"
set rs=conn.execute(Sql)
else

```

```

    Sql='select TOP 20 * from bt_tbh_kls where nim='&nim&' order
    by semester,kode_mk,kelas,sks desc'
    set rs=conn.execute(Sql)
end if
%>

```

3. *Source* saat dipilih salah satu kelas pada daftar kelas yang dibuka

```

<!--#include File="database.asp"-->
<%
nim=trim(request.QueryString("nim"))
mk2=trim(request.QueryString("kode_mk2"))
kelas2=trim(request.QueryString("kelas2"))
ip=trim(request.ServerVariables("REMOTE_ADDR"))
'bentrok
Sql_cek3='select distinct * from bt_12_8_sm where kode_mk like
"&mk2&"%' and kelas="&kelas2&'"'
set rs_cek3=conn.execute(Sql_cek3)
while not rs_cek3.eof
    Sql_cek4='select * from view_bt_kst2 where nim="&nim&" and
    right(No_Unit,3)="&right(trim(rs_cek3("No_Unit")),3)&'
    set rs_cek4=conn.execute(Sql_cek4)
    If not rs_cek4.eof then
        response.Redirect("kst.asp?nim="&nim)
    end if
rs_cek3.movenext
wend

'looping kelas jika ada kelas praktikumnya
Sql_kelas1='select * from BT_12_8_sm where kode_mk like
"&mk2&"%' and kelas="&kelas2&" and jam=1"
set rs_kelas1=conn.execute(Sql_kelas1)
while not rs_kelas1.eof
    'kode kelas
    if right(trim(rs_kelas1("kode_mk")),1)="P" then
        kk2=left(trim(rs_kelas1("kode_mk")),5)&kel(kelas2)&"P"
    else
        kk2=trim(rs_kelas1("kode_mk"))&kel(kelas2)
    end if
end while

```

end if

*'cek jumlah sks*

If right(trim(rs\_kelas1("kode\_mk")),1) <> "P" then

sk=0

Sql\_sks="select distinct \* from View\_BT\_KST2 where  
nim=" &nim&" and jam=1 and sks <> 0"

set rs\_sks=conn.execute(Sql\_sks)

while not rs\_sks.eof

sk=int(sk)+int(rs\_sks("sks"))

rs\_sks.movenext

wend

jum\_sks\_new=int(sk)+int(rs\_kelas1("sks"))

If int(jum\_sks\_new) > 20 then

response.Redirect("kst.asp?nim=" &nim)

end if

end if

*'cek jumlah mhs dan quota*

If right(trim(rs\_kelas1("kode\_mk")),1) <> "P" and

trim(rs\_kelas1("kode\_mk")) <> "TI999" and

trim(rs\_kelas1("kode\_mk")) <> "TA100" and

trim(rs\_kelas1("kode\_mk")) <> "TA101" and

trim(rs\_kelas1("kode\_mk")) <> "KP100" then

Sql\_cek="select \* from BT\_14\_1 where  
kode\_kelas=" &kk2&"

set rs\_cek=conn.execute(Sql\_cek)

Sql\_cek2="select \* from CT\_Quota where  
kode\_kelas=" &kk2&"

set rs\_cek2=conn.execute(Sql\_cek2)

If int(rs\_cek("Jum\_Mhs")) >= int(rs\_cek2("Maksimal")) then

response.Redirect("kst.asp?nim=" &nim)

end if

end if

*'apa ada di KST*

Sql\_cek5="select \* from view\_bt\_kst2 where nim=" &nim&" and  
kode\_mk=" &trim(rs\_kelas1("kode\_mk"))&"

```

set rs_cek5=conn.execute(Sql_cek5)
If not rs_cek5.eof then
    response.Redirect("kst.asp?nim=" & nim)
end if

```

```

sqla="select * from BT_12_8_sm where kelas=" & kelas2 & " and
kode_mk=" & trim(rs_kelas1("kode_mk")) & " and jam=1"
set rsa=conn.execute(sqla)
sqlb="select * from BT_12_8_sm where kelas=" & kelas2 & " and
kode_mk=" & trim(rs_kelas1("kode_mk")) & " and jam=2"
set rsb=conn.execute(sqlb)
if ucase(rsa("hari"))=ucase("senin") then
    urut=1
elseif ucase(rsa("hari"))=ucase("selasa") then
    urut=2
elseif ucase(rsa("hari"))=ucase("rabu") then
    urut=3
elseif ucase(rsa("hari"))=ucase("kamis") then
    urut=4
elseif ucase(rsa("hari"))=ucase("jumat") then
    urut=5
elseif ucase(rsa("hari"))=ucase("sabtu") then
    urut=6
end if

```

*'update table KST*

```

sql2a="insert into view_bt_kst2
(akhir,hari,jam,kelas,kode_mk,mulai,nim,ruang,shift,sks,urut,No_Unit)
values (" & rsa("akhir") & ", " & rsa("hari") & ", " & rsa("jam") & ",
" & rsa("kelas") & ", " & rsa("kode_mk") & ", " & rsa("mulai") & ",
" & nim & ", " & rsa("ruang") & ", " & rsa("shift") & ", " & rsa("sks") & ",
" & urut & ", " & rsa("No_Unit") & ")")
set rs2a=conn.execute(sql2a)
sql2b="insert into view_bt_kst2
(akhir,hari,jam,kelas,kode_mk,mulai,nim,ruang,shift,sks,urut,No_Unit)
values (" & rsb("akhir") & ", " & rsb("hari") & ", " & rsb("jam") & ",
" & rsb("kelas") & ", " & rsb("kode_mk") & ", " & rsb("mulai") & ",

```

```

""&nim&", ""&rsb("ruang")&", ""&rsb("shift")&", ""&rsb("sks")&",
""&urut&", ""&rsb("No_Unit")&""))"
set rs2b=conn.execute(sql2b)

```

```

'update table penghuni kelas
sql4="insert into BT_16_1 (kelas,kode_kelas,kode_mk,nim,shift) values
("&kelas2&", ""&kk2&", ""&trim(rs_kelas1("kode_mk"))&",
""&nim&", ""&rsa("shift")&""))"
set rs4=conn.execute(sql4)

```

```

'update table populasi
sql6="select jum_mhs from BT_14_1 where kode_kelas=""&kk2&"""
set rs6=conn.execute(sql6)
    jum2=rs6("jum_mhs")+1

```

```

'update jum_mhs untuk kk2
sql13a="update BT_14_1 set jum_mhs=""&jum2&" where
kode_kelas=""&kk2&"""
conn.execute(sql13a)

```

```

'update tabel bentrok
Sql9="select * from View_BT_KST2 where NIM=""&nim&"" and
Jam=1"
set rs9=conn.execute(Sql9)
JumKelas=1
while not rs9.eof
    sql10="select * from View_BT_KST2 where NIM=""&nim&"" and
kode_MK<>""&rs9("Kode_MK")&"" and Hari=""&rs9("Hari")&""
and Mulai=""&cdate(hour(rs9("Mulai"))&":"&minute(rs9
("Mulai"))&":"&second(rs9("Mulai"))&""))&"""
    set rs10=conn.execute(sql10)
    while not rs10.eof
        if not rs10.eof then
            JumKelas=JumKelas+1
        end if
        rs10.movenext
    wend
rs9.movenext

```

```

wend
if jumkelas>1 then
    sql11="insert into BT_16_1_1 (nim) values (“&nim&”)"
    set rs11=conn.execute(sql11)
end if

rs_kelas1.movenext
wend

'bt_log_jml_proses untuk menyatakan bahwa sudah adanya
perubahan data dan DMQ tidak upate lagi
Sql_proses1="select * from sysobjects where xtype='U' and
Name='bt_log_jml_proses'"
set rs_proses1=conn.execute(Sql_proses1)
if rs_proses1.eof then
    Sql_create_proses="create table bt_log_jml_proses (Tgl datetime,
NIM varchar(10),Jml int)"
    set rs_create_proses=conn.execute(Sql_create_proses)
    Sql_proses2="insert into bt_log_jml_proses (Tgl,NIM,Jml)
values (“&date()&” ,”&nim&” ,1)"
    set rs_proses2=conn.execute(Sql_proses2)
else
    Sql_proses3="select Jml from bt_log_jml_proses where
nim=’&nim&’"
    set rs_proses3=conn.execute(Sql_proses3)
    if rs_proses3.eof then
        Sql_proses4="insert into bt_log_jml_proses (Tgl,NIM,Jml)
values (“&date()&” ,”&nim&” ,1)"
        set rs_proses4=conn.execute(Sql_proses4)
    else
        Jml=int(rs_proses3("Jml"))+1
        Sql_proses4="update bt_log_jml_proses set Jml=’&Jml&’
where nim=’&nim&’"
        set rs_proses4=conn.execute(Sql_proses4)
    end if
end if
response.Redirect("kst.asp?nim="&nim)
%>

```

## KESIMPULAN

*Write Validation* merupakan salah satu bagian penting pada proses pembuatan Otomatisasi Online Jadwal Rencana Study (O-OJRS). Cara kerjanya yang selektif mengontrol segala kondisi sebelum suatu proses dieksekusi, membuat data dan sistem menjadi aman walaupun diakses oleh berbagai user di berbagai client. Namun pengecekan tersebut tetap memperhitungkan standar TWT (*Tolerable Wait Time*) agar tidak lebih dari 8,6 detik display data. Sehingga kenyamanan pengguna terpenuhi, dan keamanan data pun tetap terjaga.

## DAFTAR PUSTAKA

- [1] Filipova Nadezhda dan Filipov Filcho. *Development Of Database For Distributed Information Measurement And Control System* University of Economics. Varna, Bul. Kniaz Boris I. 2008.
- [2] Desmet Lieven, Jacobs Bart, Piessens Frank, Joosen Wouter. *A Generic Architecture for Web Applications to Support Threat Analysis of Infrastructural Components*. Eighth IFIP TC-6 TC-11 Conference on Communications and Multimedia Security (CMS). 2004.
- [3] Lin. J. L., Dunham M. H. and Nascimento M. A. *A Survey of Distributed Database Checkpointing*. Texas: Department of computer science and engineering, Shouthern Methodist University. 1997.
- [4] DeWitt. D.J., Gray.J. *Parallel Database Systems: The Future of High Performance Database Processing*. San Francisco: Computer Sciences Department, University of Wisconsin. 1992.
- [5] Steven P Coy. *Security Implications of the Choice of Distributed Database Management System Model: Relational Vs Object Oriented*. University of Maryland. 2008.
- [6] Stephane Gangarski, Claudia Leon, Hurbert Naacke, Marta Rukoz and Pablo Santini. *Integrity Constraint Checking In Distributed Nested Transactions Over A Database Clustur*. Laboratorie the Information Paris 6. University Pierre et Marie Curie 8 rue du Capitaine Scott, 75015, Paris. Centro de Computacion Paralela Y Distribuida, Universidad Central de Venezuela Apdo. 47002, Los Chaguaramos, 1041 A, Caracas, Venezuela. 2006.
- [7] Stanchev Lubomir. *Semantic Data Control In Distributed Database Environment*. University of Waterloo. 2001.

- [8] Powell James C .Allison L, dkk. *The Impact of Database Selection on Distributed Searching* Perancis Departemen Ilmu Komputer Universitas Virginia.
- [9] S. Aljawarneh, and F. Alkhateeb, A semantic web technology-based architecture for new server - side data validation in web applications, in *Proceedings of International Conference on Information Technology*, Amman Jordan, 2009, pp. 155-160.
- [10] Ibrahim Hamidah. *Deriving Global Integritas Dan Local Rules For Distributed Database* Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Putra Malaysia, 43400 UPM Serdang.
- [11] Rahardja Untung and Badar Shakinah. *Penerapan metode Data Mart Query (DMQ) dalam Distributed Database System*. Perguruan Tinggi Raharja of Indonesian. 2009.
- [12] W.B. Glisson and R. Welland, Web development evolution: The assimilation of web engineering security, in *Proceedings of the Third Latin American Web Congress*, Washington, DC, USA, IEEE Computer Society, 2005, pp. 49-55.
- [13] S. Sedaghat, J. Pieprzyk, and E. Vossough, On the web content integrity check boosts users' confidence, *ACM Communication*, vol. 45, no. 11, pp. 33-37, 2002.
- [14] Y. Huang, S. Huang, T. Lin, and C. Tsai, Web application security assessment by fault injection and behavior monitoring, in *Proceedings of the 12th international conference on World Wide Web*, New York, NY, USA, ACM Press , 2003, pp. 148-159.
- [15] Website Optimization (2010). Web Page Performance Thesis. Diakses pada tanggal 24 mei 2010 dari : <http://www.websiteoptimization.com/speed/tweak/web-page-performance-thesis/>