PENDEKATAN *ACTIVE DATABASE SYSTEM* DAN *BUSINESS RULE* DALAM PENGEMBANGAN SISTEM INFORMASI

[1]M. Miftakul Amin, [2]Mustaziri

[1, 2] Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya Palembang
Jalan Srijaya Negara, Palembang 30139
Telp. 0711 – 353414 Fax. 0711 – 355918
e-mail : mafis_amin@mail.ugm.ac.id

ABSTRACT

An information system is inseparable with the database as data repository and give the data service to information system. Information system basically translate the business rule from an organization into form of algorithm programming which later then planted in application. Database which used conventionally have the character of passive and there is no controlling mechanism, that happened only mechanism just enter and its exit of data. With the existence of active of database system, a database earn the reactive behavior and conduct the controlling of information system at database level. Business rules which is as a rule planted in application can be planted directly into database and can be used concurrently by entire/all application that using database. This writing give the study about an alternative approach of business rules and active database system in developing an information system by planting business rules in the form of programming algorithm into database management system so that more improving the performance of information system. This active database system approach can be implemented not only in the field of information systems development, but can be used as a data base responsive to collaborate with the control system hardware.

Key words : *information system, active database system, business rule*

ABSTRAK

Sebuah sistem informasi tidak dapat dipisahkan dengan database sebagai tempat penyimpanan data dan memberikan layanan data ke sistem informasi. Sistem informasi pada dasarnya menerjemahkan aturan bisnis dari sebuah organisasi ke dalam bentuk program algoritma yang kemudian kemudian ditanam dalam aplikasi. Database yang digunakan secara konvensional bersifat pasif dan tidak ada mekanisme pengendalian, yang terjadi hanya mekanisme masuk dan keluarnya data saja. Dengan adanya *active database system*, database mendapatkan perilaku reaktif dan melakukan pengendalian sistem informasi pada tingkat database . Aturan bisnis yang berfungsi sebagai aturan ditanam di aplikasi dapat ditanam langsung ke dalam database dan dapat digunakan secara bersamaan oleh seluruh / semua aplikasi yang menggunakan basis data . Tulisan ini memberikan studi tentang pendekatan alternatif aturan bisnis dan *active database system* dalam mengembangkan suatu sistem informasi dengan menanam aturan bisnis dalam bentuk algoritma pemrograman ke dalam sistem manajemen database sehingga lebih meningkatkan kinerja sistem informasi. Pendekatan *active database system* ini dapat diimplementasikan tidak hanya pada bidang pengembangan sistem informasi, tetapi dapat digunakan sebagai basis data yang responsif untuk berkolaborasi dengan sistem kendali pada perangkat keras.

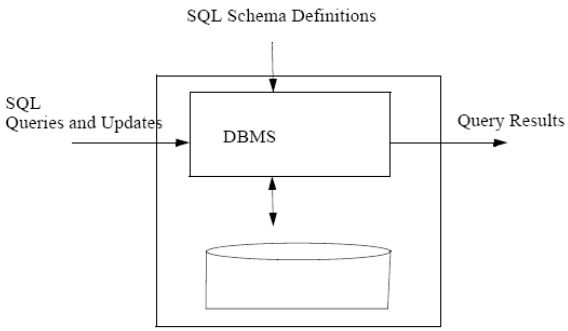
Kata kunci : *sistem informasi, active database system, business rule*

I. PENDAHULUAN

Sebuah Sistem Informasi tidak dapat dipisahkan dari aplikasi dan basisdata sebagai tempat penyimpanan data. Basisdata merupakan salah satu elemen penting yang tidak dapat dipisahkan dari sebuah sistem informasi. Basisdata dapat diibaratkan sebagai sebuah pondasi bagi sistem informasi untuk menyediakan informasi dan data yang dibutuhkan, baik yang digunakan untuk menunjang sistem itu sendiri ataupun sebagai output untuk pengguna [1].

Saat ini terjadi perubahan yang cukup pesat secara global, sehingga sebuah perusahaan yang sukses harus didukung oleh banyaknya proses pengambilan keputusan. Hal ini tidak hanya diartikan sebagai mengumpulkan dan memproses data menggunakan sistem informasi, tetapi juga membuat keputusan dengan dukungan aturan-aturan bisnis. Aturan bisnis diartikan sebagai pernyataan tentang bagaimana suatu proses bisnis dilakukan. Mengolah aturan bisnis dalam media penyimpanan menjadi hal yang krusial dalam semua sistem informasi, karena aturan bisnis menjadi sebuah pengetahuan (*knowledge*) sebagai asset utama bagi perusahaan. Analis bisnis dapat dengan cepat membuat keputusan, jika aturan bisnis dibuat otomatis. Proses otomatisasi ini membutuhkan lingkungan yang menyediakan sumber daya untuk mengelola dan menyimpan aturan tersebut [2].

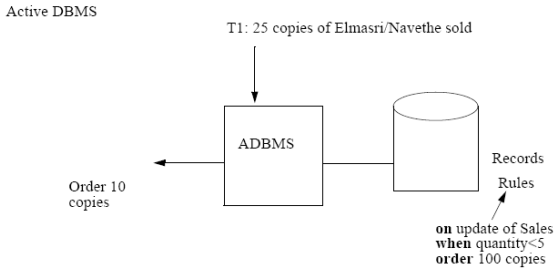
Sebuah basis data konvensional hanya mempunyai fungsi untuk melayani data masuk dan keluar dalam sebuah sistem informasi, tanpa memiliki fungsi tambahan untuk berperilaku reaktif dan meresponse terhadap lingkungan luar basisdata. Proses pengontrolan data dilakukan langsung dari client. Model database konvensional dapat dilihat pada gambar 1.



Gambar 1. Model Database Konvensional

Sedangkan pada model *active database system*, sebuah perangkat lunak basisdata mampu mengenali perubahan yang terjadi di lingkungan luar database.

Active database system menggunakan *predefined rule* (aturan-aturan yang telah didefinisikan sebelumnya) yang ditanam dalam database. Setiap kali terjadi perubahan lingkungan basisdata, maka terdapat aturan yang akan mengontrol perubahan tersebut. Model Active Database dapat dilihat pada gambar 2.



Gambar 2. Model Active Database

Basisdata adalah sebuah koleksi dari data yang tahan lama yang digunakan oleh sistem informasi atau aplikasi dari perusahaan tertentu. Istilah perusahaan di sini hanyalah istilah yang memudahkan untuk organisasi yang cukup komersial, ilmiah, teknis, atau lainnya. Sebuah perusahaan dapat merupakan individual (dengan sebuah basisdata perorangan yang kecil), atau gabungan lengkap atau badan usaha besar yang serupa [6].

Secara konvensional sebuah sistem basisdata bersifat pasif, menyimpan dan menampilkan secara langsung terhadap response yang diberikan oleh user tanpa ada operasi spesifik dalam basisdata tersebut. *Active Database Management System* (ADBMS) pada dasarnya adalah sebuah database konvensional yang sifatnya pasif, dengan kemungkinan untuk berperilaku secara reaktif. Penambahan fungsional reaktif ini ditandai dengan adanya ECA-rules (event-condition-action rules) yang diartikan dengan "jika sebuah event terjadi, cek terhadap kondisi, dan jika bernilai benar, maka sebuah aksi akan dilaksanakan". Sekali sekelompok aturan-aturan didefinisikan, sebuah *Active Database Management System* akan melakukan monitoring terhadap event yang terjadi [7].

Sebuah perangkat lunak manajemen database dapat dikategorikan ke dalam *Active Database Management System* jika di dalam perangkat lunak tersebut memiliki fitur sebagai berikut [7]:

1. Fasilitas mendefinisikan ECA-Rule
   Sebuah active database pada dasarnya merupakan sebuah *database management system* konvensional. Kemudian ditambahkan fitur reaktif

dengan disediakannya fungsionalitas ECA-Rule.

2. Mempunyai model eksekusi.
   Sebuah *Active Database Management System* memiliki kemampuan untuk mendeteksi sebuah kejadian yang memicu aturan untuk dijalankan.
3. Mendukung Pemakaian dan Aplikasi.
   Sebuah *Active Database Management System* mendukung lingkungan pemrograman, artinya dapat diintegrasikan dengan development tools untuk membuat sebuah aplikasi. ADBMS menyediakan fasilitas untuk melihat aturan, mendefinisikan aturan, dan melakukan perbaikan jika terjadi error pada sebuah rule.

Business Rule atau Aturan bisnis merupakan kalimat yang mendefinisikan atau membatasi beberapa aspek bisnis. Aturan bisnis dimaksudkan untuk menjamin struktur bisnis atau mengendalikan perilaku bisnis. Saat ini kebanyakan organisasi dituntun oleh ratusan, bahkan ribuan kombinasi aturan bisnis. Agregasi dari aturan bisnis yang diterapkan berpengaruh terhadap perilaku dan menentukan cara organisasi menanggapi lingkungan [8].

Konsep aturan bisnis telah digunakan di sistem informasi dalam waktu yang cukup lama. Namun, lebih sering istilah ini digunakan berhubungan dengan istilah konstrain integritas. Tetapi konstrain integritas lebih sempit cakupannya dibanding dengan aturan bisnis, karena konstrain integritas hanya mengacu pada pemeliharaan nilai-nilai data absah di basisdata.

Pendekatan aturan bisnis dapat digunakan untuk menspesifikasikan kebutuhan sistem informasi berdasarkan alasan-asalan berikut:

1. Aturan bisnis adalah konsep inti di perusahaan karena merupakan ekspresi dari kebijakan bisnis dan menuntut perilaku individu dan agregasinya.
2. Aturan bisnis dapat dinyatakan dalam istilah-istilah yang akrab pada pemakai akhir. Dengan demikian, pemakai akhir dapat mendefinisikan dan memelihara aturan-aturannya sendiri.
3. Aturan bisnis sangat terpelihara. Aturan-aturan tersebut tersimpan dalam repositori pusat dan masing-masing aturan diekspresikan hanya sekali kemudian dapat digunakan bersama di seluruh elemen organisasi.

Pemaksaan aturan bisnis dapat diotomatisasikan lewat penggunaan perangkat lunak yang dapat menginterpretasikan aturan dan memaksakannya menggunakan

mekanisme integritas dari sistem manajemen basisdata.

Aturan bisnis diartikan sebagai pernyataan yang mendefinisikan atau membatasi beberapa aspek bisnis [9]. Aturan bisnis diklasifikasikan menjadi:

1. Strucutral Assertion, aturan yang mengungkapkan struktur statis (tetap). Aturan ini adalah kalimat yang menggambarkan beberapa konsep bisnis yang berhubungan dengan struktur sebuah organisasi.
2. Derivation, aturan yang diturunkan dari pengetahuan lain.
3. Action Assertion, aturan yang mengungkapkan batasan/kendali dari aspek dinamis sebuah perusahaan. *Action assertion* dapat diimplementasikan ke dalam sebuah database management system menggunakan *trigger* dan *stored procedures.*

Aturan bisnis lain yang juga dapat diterapkan dalam active database system [2], yaitu :

1. Stimulus/*Response Rules*, yaitu sebuah aturan menggambarkan tindakan yang harus dieksekusi sebagai response untuk menjawab kejadian dari event yang terjadi sebelumnya.
2. Operational Rules, yaitu sebuah aturan yang menggambarkan suatu tindakan atau suatu urutan operasi yang harus dilakukan untuk mencapai beberapa tujuan.

Sesungguhnya yang dimaksud dengan sistem informasi tidak harus melibatkan komputer. Sistem informasi yang melibatkan komputer disebut sistem informasi berbasis komputer (*Computer based information system* atau CBIS). sebuah sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang diproses (data menjadi informasi), dan dimaksudkan untuk mencapai suatu sasaran atau tujuan [10].

Menurut McFadden (1999) dalam Kadir (1993) mendefinisikan informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut [10]. Menurut Alter (1992) dalam Kadir (1993) mendefinisikan pengetahuan (knowledge) adalah kombinasi dari naluri, gagasan, aturan, dan prosedur yang mengarakan tindakan atau keputusan. Sebagai gambaran informasi yang dipadukan dengan pengalaman masa lalu dan keahlian akan memberikan suatu pengetahuan yang tentu saja memiliki nilai yang tinggi [10].

II. METODE PENELITIAN

Ketika mengembangkan sebuah sistem informasi dengan landasan aturan bisnis, secara fisik aturan-aturan bisnis tersebut disimpan dalam sebuah *repository* atau tempat penyimpanan data yang dapat dikelola dan diubah setiap saat menggunakan perangkat lunak yang tersedia. Ada 2 solusi yang dapat digunakan dalam mengembangkan sistem informasi [3] yaitu:

1. Parameter Driven Approach
   Dalam pendekatan ini aturan-aturan disimpan dalam sebuah perangkat lunak basisdata. Aturan-aturan tersebut dikenali dengan sejumlah atribut atau nilai berupa parameter sehingga mampu berkomunikasi dengan aplikasi front-end yang menggunakan aturan. Aplikasi akan mengirimkan nilai-nilai berupa parameter yang telah didefinisikan di dalam aturan yang tersimpan di dalam database. Sehingga melalui parameter-parameter inilah aplikasi dan basisdata saling berkomunikasi untuk bertukar data dan informasi.
2. Independent Process Driven Approach
   Dalam pendekatan ini, aturan-aturan mengikuti pola pengembangan sistem informasi tradisional. Aturan diterjemahkan ke dalam kode program di aplikasi secara langsung. Aturan-aturan tersebut tidak disimpan di dalam database, tetapi dalam sebuah lapisan tersendiri, sehingga dapat diakses setiap saat ketika sistem membutuhkan eksekusi aturan tersebut.

Salah satu fitur yang terdapat dalam Active Database System adalah adanya mekanisme pendefinsian Event – Condition – Action (ECA) Rule. Event – Condition – Action (ECA) adalah sebuah cara yang digunakan untuk menangkap perilaku dinamis dalam sebuah sistem informasi. Paradigma ECA telah memberikan dampak yang siginifikan di bidang Sistem Informasi dan telah digunakan dalam *Active Database* baik secara konseptual maupun dalam implementasinya.

Menurut Dayal (1997, h. 1) Paradigma aturan produksi pada *Active Database Management System (ADBMS)* mengikuti pola aturan produksi pada *Artifficial Intelligence* (AI) dengan aturan menyerupai aturan produksi pada sistem pakar [11], yaitu dengan notasi:

Condition → Action

Suatu mesin kesimpulan beredar mengelilingi aturan di dalam sistem, untuk mempertemukan kondisi bagian dari aturan dengan data dalam memori kerja. Paradigma ini telah membuat sebuah pola

di dalam ADBMS terkait ECA-rule dengan pola sebagai berikut:

On event

If Condition

Then Action

Event dapat didefinisikan sebagai sesuatu yang terjadi pada titik waktu tertentu. Sumber dari event dapat dihasilkan oleh beberapa hal sebagai berikut [12]:

a. Structure Operation

Yaitu even yang dihasilkan operasi terstruktur dalam sebuah database, seperti menambahkan sebuah record, menghapus, mengupdate atau menampilkan sekumpulan record tabel.

b. Behavior Invocation

Yaitu even yang dihasilkan dari eksekusi beberapa operasi yang didefiniskan oleh user, misal pada saat kotak pesan ditampilkan atau seorang user menekan tombol OK.

c. Transaction

Yaitu sebuah even yang dihasilkan dari perintah transaksi, seperti abort, commit, dan begin-transaction.

d. Abstract atau User Defined

Yaitu even yang terjadi atas perilaku dari luar sistem, sehingga even dihasilkan secara explicit, misalnya user mengetikkan sebuah informasi ke dalam sistem.

e. Exception

Yaitu sebuah even yang dihasilkan dari beberapa excepsi dari sistem, seperti pesan ketika seseorang tidak mempunyai wewenang untuk mengakses record di dalam tabel.

f. Clock

Yaitu even yang dihasilkan karena beberapa titik waktu didefiniskan, misalnya even terjadi ketika pada tanggal tertentu, hari tertentu atau pukul tertentu.

g. External

Yaitu even yang terjadi karena faktor dari luar database, seperti misalnya suhu mencapai lebih dari 30 derajat.

Dasar bagi pemicu terjadinya *event* adalah perintah standar manipulasi data SQL, yaitu perintah INSERT, UPDATE dan DELETE. Perintah standar SQL tersebut membentuk sebuah konsep nilai OLD dan NEW. Nilai NEW diartikan sebagai nilai yang diperoleh dari record akibat perintah INSERT dan DELETE. Dan nilai OLD diartikan sebagai nilai yang telah dihapus akibat perintah DELETE atau nilai sebelum dikenai perintah UPDATE [4].

Condition dapat diartikan sebagai sesuatu yang mempunyai peran kapan sesuatu itu akan dilaksanakan. Condition dievaluasi dengan nilai boolean True atau False. Pada ECA-Rule secara umum

condition berprilaku secara *optional* artinya dapat digunakan atau tidak. *Action* merupakan cakupan tugas yang dibuat sesuai dengan aksi secara spesifik. Aksi dapat berupa tugas tertentu sesuai dengan pendefinisiannya [12].

Action dari konsep ECA-rule dapat berupa pernyataan SQL untuk memanipulasi data, juga dapat berupa prosedur yang dibuat untuk memproses data, serta pemanggilan terhadap sebuah *stored procedure* yang sebelumnya telah didefinisikan.

III. HASIL DAN PEMBAHASAN

Dalam implementasinya ECA-rule tidak harus memenuhi seluruh event-condition-Action. Bisa jadi ECA-rule hanya menerapkan Event dan Action (EA) demikian juga hanya menerapkan Condition untuk melakukan Action tertentu (CA). Anatomi dari tipe aturan dapat dilihat pada tabel 1.

Tabel 1. Anatomi Tipe Rule

Tipe Rule	**Anatomi**
Derivation Rule	Condition – Action (CA) Condition Primary Action – Secondary Action (CA | CA2)
Action Rule	Action (A)
Stimulus/Response	Event – Condition – Action (ECA) Event – Action (EA) Event – Condition – Primary Action – Secondary Action (ECA | A2)

Active Rule merupakan fungsionalitas tambahan yang ada pada perangkat lunak basisdata. Mekanisme aktif ini didasarkan pada aturan *event-condition-action* (ECA) rule. Aturan-aturan tersebut secara otomatis dipicu berdasarkan *event* yang terjadi. Pada beberapa database komersial fungsionalitas *active rule* diimplementasikan dalam bentuk trigger [2].

Active Rule atau aturan aktif pada dasarnya adalah aturan-aturan yang disimpan dalam sebuah *Active Database System*. Secara umum aturan dipicu oleh adanya aktifitas dalam sistem, misalnya karena ada pengaruh penambahan baris/record pada tabel dan penghapusan atau modifikasi record.

Jika aturan bisnis diimplementasikan dalam sistem informasi, menggunakan sebuah bahasa pemrograman, maka kode program yang merupakan implementasi dari aturan bisnis tersebut tersebar ke dalam beberapa aplikasi. Jika suatu saat aturan bisnis berubah, maka pengembang sistem harus

membedah kembali aplikasi yang telah dibuat untuk menemukan aturan terkait yang kemudian diubah, dan dikompile ulang. Proses modifikasi seperti ini adalah hal yang sulit dan memakan waktu. Di sisi lain, ketika aturan-aturan tersebut di implementasikan dalam sebuah *Active Database System*, aturan-aturan tersebut dibuat sekali, kemudian dikelola secara terpusat oleh *Active Database Management System (ADBMS)*, disimpan dalam sebuah media penyimpanan aturan, dan dipakai secara bersamaan oleh semua aplikasi atau progam yang mengakses database tersebut.

Sebuah sistem informasi terdiri dari manajemen data dan presentation (end user), ketika sistem informasi menerapkan aturan bisnis dalam pengembangan sistem informasi, maka terdapat tiga lapisan yaitu manajemen data, presentasion (end user) dan lapisan aturan bisnis. Aturan-aturan bisnis tersebut disimpan dalam sebuah penyimpanan (repository) dalam perangkat lunak basisdata [3].

Dalam sebuah *Active Database System* terdapat fungsionalitas tambahan untuk membuat aturan-aturan aktif (*active rules*). Aturan-aturan tersebut dapat secara otomatis dipicu berdasarkan kejadian yang spesifik yang terjadi dalam basisdata. Beberapa perangkat lunak sistem manajemen basisdata telah menambahkan fungsionalitas untuk mendukung pemakaian *active rules* dengan fitur yang disebut *trigger*[4]. Sebuah *Active Database Management System* (ADBMS) mendukung pembuatan, pengolahan dan eksekusi Event-Condition-Action (ECA) rule.

Trigger merupakan sebuah prosedur yang secara otomatis dijalankan oleh perangkat lunak sistem manajemen basisdata (DBMS) dalam merespon perubahan yang terjadi dalam database. Sebuah database yang memiliki kumpulan *trigger* ini disebut *Active Database System*. Trigger tersebut berperan sebagai *daemon* yang setiap saat memonitoring aktifitas database [5].

Aturan bisnis diimplementasikan ke dalam bentuk *trigger* ketika aturan-aturan bisnis tersebut relatif kompleks. Keuntungan dari penggunaan *trigger* adalah mampu mengekspresikan algoritma yang kompleks yang biasa dijumpai pada setiap aplikasi program [2].

IV. KESIMPULAN

1. Pengembangan sebuah sistem informasi menggunakan active database system sebagai tempat penyimpanan aturan-aturan bisnis dalam sistem informasi dapat digunakan sebagai alternatif dalam menggantikan algoritma pemrograman

dari aturan bisnis yang biasa ditanam dalam kode program.

2. Komunikasi antara *active database system* dan aplikasi dapat menggunakan model pendekatan parameter atau membuat layar di aplikasi yang menyimpan aturan bisnis, sehingga dapat di-*share* antar aplikasi yang menggunakan database.
3. Active *Database System* dapat menggunakan *Trigger*, *Rule*, *Stored Procedure* yang saat ini telah disediakan oleh beberapa database server yang ada di pasaran.

FUTURE WORKS

Penulisan ini memberikan paparan dalam tataran konsep yang mengusulkan sebuah model pengembangan sistem informasi dengan memetakan sebagian aturan bisnis yang selama ini ditanam dalam aplikasi kemudian ditulis dalam lapisan *database managemement database system* (DBMS). Penelitian lanjutan dapat melakukan implementasi menggunakan beberapa DBMS yang mendukung pembuatan *rule*/aturan bisnis dalam fitur yang melekat pada DBMS. Setelah proses implementasi dapat dilanjutkan dengan membuat perbandingan kelebihan dan kekurangan yang muncul jika diimplementasikan ke dalam masing-masing DBMS.

ACKNOWLEDGMENT

Ucapan terimakasih kepada seluruh teman sejawat di Jurusan Teknik Komputer Politeknik Negeri Sriwijaya Palembang, juga kepada seluruh civitas akademika IIB darmajaya Bandar Lampung terutama kepada lembaga penelitian yang telah memberikan kesempatan kepada penulis sehingga naskah ini dimuat dalam jurnal Informatika.

DAFTAR PUSTAKA

[1] Adriansyah. 2005. “Implementasi Basisdata Dalam Real-Time System”. Program Studi Teknik Informatika Institut Teknologi Bandung.

[2]. Viana, S.; Junior J.R.d.A; Pavon, J. 2007. “A Rule Repository for Active Database System”. *Cley Electronic Jurnal*, Volume 10, Number 2, Paper 4.

[3]. Butleris, R; Kapocius, K. 2002. “The Business Rules Repository For Information Systems Design”. ADBIS. Department of Information Systems, Kaunas University of Technology, Lithuania.

[4]. Elmashri, R., Navathe, B. 2000. “Fundamentals of Database Systems, Third Edition”, Addison Wisley.

[5]. Ramakrishnan, R.; Gehrke, J. 2007. "Database Management Systems, Second Edition". Mc. Graw-Hill.

[6]. Date, C.J. 2000. "An Introduction to Database Systems". Addison Wesley. New Jersey.

[7]. ACT-NET Consortium. 1996. "The Active Database Management System Manifesto: A Rulebase of ADBMS Features". *ACM SIGMOD Record*, v.25, n.3, p.40-49.

[8]. Hariyanto, B. 2004. "Sistem Manajemen Basisdata, Pemodelan, Perancangan dan Terapannya". Penerbit Informatika, Bandung.

[9]. Hoffer, Jeffrey, A, et all. 2007. "Modern Database Management, Eight Edition", New Jersey : Pearson Education, inc.

[10]. Kadir, A.; Triwahyuni, T.C. 2003. "Pengenalan Teknologi Informasi". Andi Offset Yogyakara.

[11]. Dayal, U.; Hanson, E., N.; Widom, J. 1994. "Active Database Systems: Modern Database System: Interoperability, and Beyond". Addison Weley, Reading, Massachusetts.

[12]. PATON, N. W.; DÍAZ, O.; 1999. "Active Database Systems". *ACM Computing Surveys*, v.31, n.1,p.63-103.