

JURNAL ILMIAH

DASI

DATA EKONOMI, BISNIS DAN TEKNOLOGI INFORMASI

AKADEMI MANAJEMEN INFORMATIKA DAN KOMPUTER
"AMIKOM" YOGYAKARTA

JURNAL ILMIAH

DASI

Aplikasi Basis Data Three-Tier

Oleh : Arief Setyanto

PENDAHULUAN

Makalah akan membahas pengembangan sistem sistem informasi akademik yang pada mulanya berbasis LAN. Untuk memberikan kemudahan kepada para pemakainya kami mencoba mengembangkan kemungkinan baru untuk dilakukan pengaksesan melalui WEB dengan menghubungkan database yang telah tersedia dengan web server.

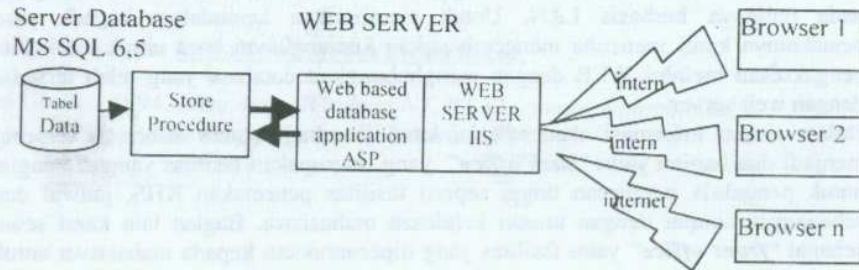
Dalam sistem informasi akademik ini kami membagi sistem informasi tersebut menjadi dua bagian yaitu "*back office*" yang merupakan fasilitas yang di bangun untuk pengelola perguruan tinggi seperti fasilitas pencetakan KHS, jadwal dan sebagainya sampai dengan urusan kelulusan mahasiswa. Bagian lain kami sebut sebagai "*front office*" yaitu fasilitas yang diperuntukkan kepada mahasiswa untuk melakukan KRS, maupun mengakses pengumuman nilai. Pada awalnya fasilitas "*front office*" tersebut di disain untuk di akses dalam jaringan lokal dengan model client server (2-tier) saja. Saat ini kami berusaha untuk mengembangkan "*front office*" menjadi web based application dengan model 3-tier dengan tetap memanfaatkan database yang telah ada.

Model pengaksesan data yang kami lakukan adalah dengan memanfaatkan metode pengaksesan data terbaru dari microsoft yaitu menggunakan ADO (Active data object). Kami tidak menggunakan DAO karena kebetulan database yang telah tersedia sudah berada di server database MS-SQL Server versi 6.5. Dimana pada server itu juga kami meletakkan database yang saat ini di pakai untuk Sistem Informasi Akademik yang sudah ada.

Untuk menjamin keamanan data sebagai aset yang sangat berharga bagi institusi penggunaannya maka kami menggunakan ASP sebagai bahasa pemrograman di WEB, di bantu dengan beberapa script java untuk mengurus tampilah di browser. ASP dapat diletakkan pada aras server (*server side*) dan tidak dapat diakses kodenya dari browser user sehingga memungkinkan kami menyembunyikan password untuk masuk ke server database. Disamping itu demi keamanan dan kecepatan aplikasi yang dijalankan di web server yang di bangun menggunakan ASP tadi juga kami buat user yang tidak diberi hak untuk mengakses table pada server database tetapi hanya berhak menjalankan (*execute*) terhadap beberapa store procedure pada server database. Hal ini merupakan upaya kami menjamin keamanan data agar seandainya saja ada user yang dengan kode kode tertentu dapat *hack* web server yang kami bangun dengan IIS milik microsoft, dan berhasil mendapatkan kode ASP untuk aplikasi WEB server, masih tidak dapat melakukan apa apa terhadap data, karena di dalam aplikasi tersebut hanya diberi

user name dan pasword yang memang didisain untuk tidak dapat melakukan apa apa.

Untuk lebih memberikan gambaran yang jelas dapat kita lihat pada gambar dibawah ini :

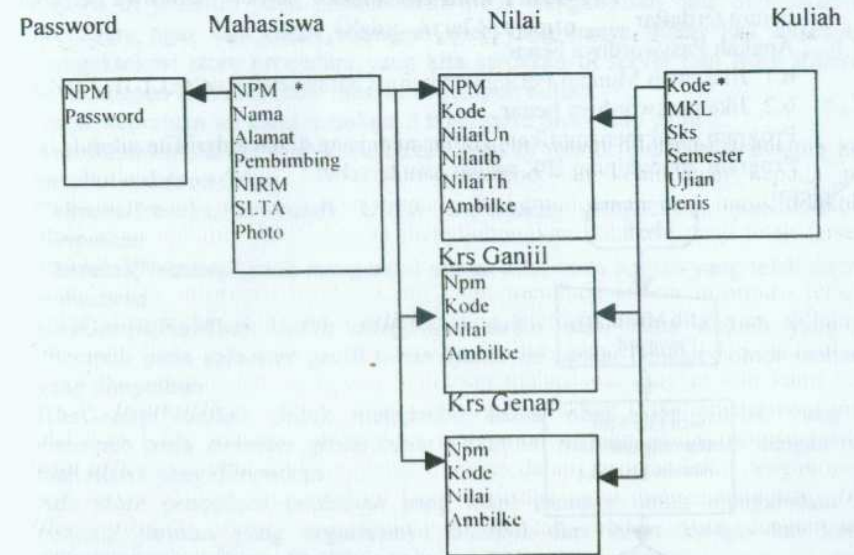


Gambar 1 Skema Sistem

Dari gambar 1 diatas dapat di lihat bahwa data yang akan diakses sangat terlindungi karena data yang tampil pada jendela browser merupakan data yang telah melewati WEB server, aplikasi pada yang dibuat dengan ASP, store procedure yang di buat dengan bahasa SQL dan baru store procedurelah yang dapat mengakses langsung ke tabel tabel yang berisi data.

STRUKTUR DATABASE

Di dalam pembahasan ini kami akan memaparkan struktur database yang kami rancang untuk keperluan front office saja, dan untuk keperluan menampilkan Pengumuman nilai keseluruhan yang jika dilakukan secara manual akan menjadi transkrip nilai. Dalam ERD yang kami gambarkan ini kami hanya menampilkan file file yang akan di gunakan untuk mendukung pengumuman nilai secara online.



Gambar 2. Entity Relationship Diagram

Primary key terdapat pada tabel mahasiswa dengan nama field NPM yang menjadi *foreign key* pada tabel tabel Nilai, KrsGanjil, KrsGenap dengan hubungan *one to many* serta pada file password dengan hubungan *One to one*. *Primary key* yang lain ada pada file Kuliah dengan nama field kode yang menjadi *foreign key* pada file Nilai, Krsganjil, KrsGenap dengan hubungan *one to many*. Untuk alasan keamanan file yang berisi password milik masing masing mahasiswa di pisahkan pada file tersendiri yang bernama password.

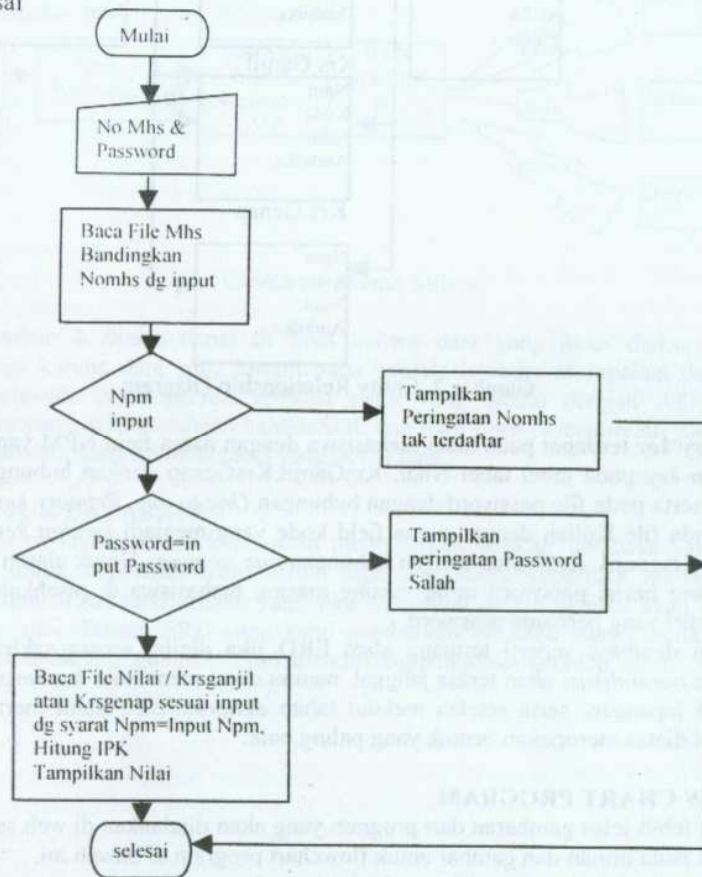
Desain database seperti tertuang alam ERD jika dinilai secara saklek menurut kaidah *normalisasi* akan terasa janggal, namun demi memenuhi *business rule* yang ada di lapangan, serta setelah melalui tahap *denormalisasi* kami merasa bentuk seperti diatas merupakan bentuk yang paling baik.

FLOW CHART PROGRAM

Untuk lebih jelas gambaran dari program yang akan dijalankan di web server dapat dilihat pada uraian dan gambar untuk flowchart program di bawah ini.

1. User Melakukan Entry Nomor Mahasiswa, Passwordnya
2. Aplikasi server akan mengecek
 - a. Apakah mahasiswa dengan nomor mahasiswa tersebut sudah terdaftar

- a.1 Jika Belum muncul Peringatan Bahwa nomor mahasiswa tersebut belum terdaftar
- b. Apakah Passwordnya benar
 - b.1 Jika salah Muncul Peringatan bahwa passwordnya salah
 - b.2 Jika passwordnya benar
 Program akan menampilkan Daftar nilai yang di baca dari file nilai
 Program menghitung IPK mahasiswa tersebut
3. Selesai



Gambar 3. Flowchart Program Untuk menampilkan pengumuman Nilai

IMPLEMENTASI

Seperti di jelaskan pada pendahuluan untuk pengaksesan data digunakan store procedure agar kita dapat membuat user yang hanya diberi hak akses untuk mengeksekusi store procedure yang kita sediakan di server tapi tidak diberi hak akses apapun terhadap table mhs, nilai ataupun kuliah.

Untuk keperluan ini kami membuat 3 buah store procedure yaitu :

Ambilnama(Nomhs) Untuk mengecek apakah nomor mahasiswa dengan nomor tersebut sudah terdaftar.

Cekpass(Nomhs,Password) Untuk melakukan pengecekan password yang diinputkan

Khstotal(Nomhs) Untuk mengambil semua nilai mata Kuliah yang telah ditempuh mahasiswa

KhsGanjil(Nomhs) Untuk mengambil daftar nilai mata Kuliah yang telah ditempuh pada semester ganjil tahun ajaran ini sesuai dengan nomor mahasiswa yang diinputkan

KhsGenap(Nomhs) Untuk mengambil daftar nilai mata Kuliah yang telah ditempuh pada semester genap tahun ajaran ini mahasiswa sesuai dengan nomor mahasiswa yang diinputkan

Ada store procedure tambahan yang kami rancang untuk mengartikan nama panjang jurusan yang argumennya diambil dari huruf ketiga dan keempat nomormahasiswa yang dituliskan yaitu **Ambiljur(Huruf34)**

Keenam store procedure ini kami simpan di server dan program yang kami buat dalam ASP tinggal memanggil saja. Dan mendapatkan hasilnya berupa data jadi misalnya begitu menjalankan cekpas maka jika hasilnya kosong berarti password tidak cocok.

Langkah yang perlu dilakukan di ASP untuk melakukan koneksi dengan database di server database dapat dilihat pada potongan kode dibawah ini:

```

strconn= "Driver={SQL
Server};SERVER=amikom32;UID=internet;PWD=AMIKOMYogyakarta;Database=amikom"
  
```

Pernyataan ini jika dieksekusi akan melakukan koneksi dari program kita ke database amikom di server database dengan nama amikom32, menggunakan user name "internet" dengan password "AMIKOMYogyakarta".

Setelah koneksi berhasil langkah selanjutnya adalah melakukan pengambilan data dengan cara mengeksekusi store procedure yang telah tersedia. Caranya dengan membuat object recordset baru yang secara detail dapat dilihat dibawah ini :

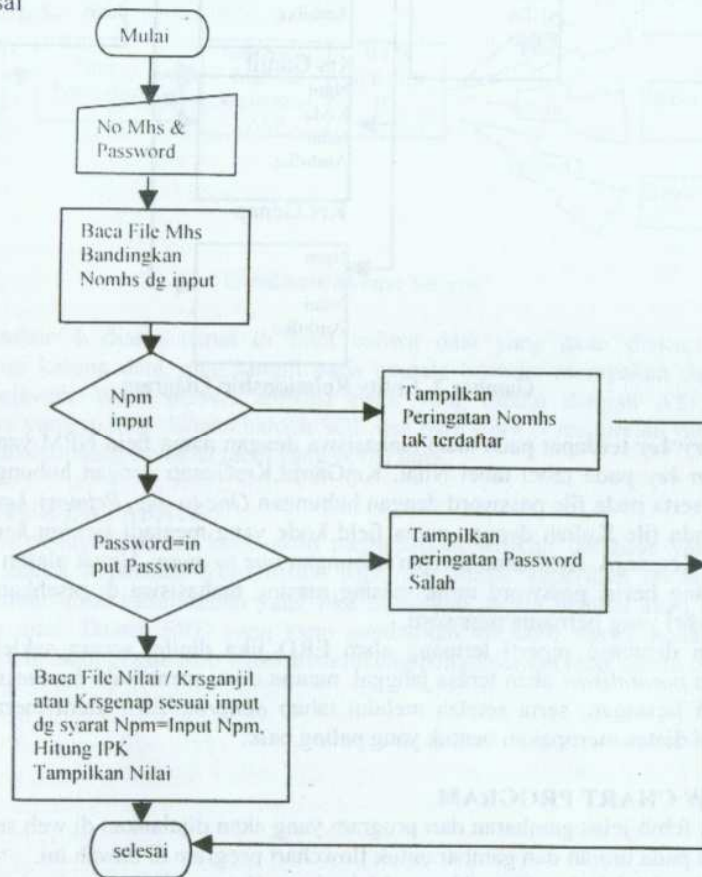
```

SQL1="Exec ambilnama '" & Request.Form(ltrim("NP"))
&"'"
  
```

```

set rsl=conn.execute(sql1)
  
```


- a.1 Jika Belum muncul Peringatan Bahwa nomor mahasiswa tersebut belum terdaftar
- b. Apakah Passwordnya benar
 - b.1 Jika salah Muncul Peringatan bahwa passwordnya salah
 - b.2 Jika passwordnya benar
 Program akan menampilkan Daftar nilai yang di baca dari file nilai
 Program menghitung IPK mahasiswa tersebut
3. Selesai



Gambar 3. Flowchart Program Untuk menampilkan pengumuman Nilai

IMPLEMENTASI

Seperti di jelaskan pada pendahuluan untuk pengaksesan data digunakan store procedure agar kita dapat membuat user yang hanya diberi hak akses untuk mengeksekusi store procedure yang kita sediakan di server tapi tidak diberi hak akses apapun terhadap table mhs, nilai ataupun kuliah.

Untuk keperluan ini kami membuat 3 buah store procedure yaitu :

Ambilnama(Nomhs) Untuk mengecek apakah nomor mahasiswa dengan nomor tersebut sudah terdaftar.

Cekpass(Nomhs,Password) Untuk melakukan pengecekan password yang diinputkan

Khstotal(Nomhs) Untuk mengambil semua nilai mata Kuliah yang telah ditempuh mahasiswa

KhsGanjil(Nomhs) Untuk mengambil daftar nilai mata Kuliah yang telah ditempuh pada semester ganjil tahun ajaran ini sesuai dengan nomor mahasiswa yang diinputkan

KhsGenap(Nomhs) Untuk mengambil daftar nilai mata Kuliah yang telah ditempuh pada semester genap tahun ajaran ini mahasiswa sesuai dengan nomor mahasiswa yang diinputkan

Ada store procedure tambahan yang kami rancang untuk mengartikan nama panjang jurusan yang argumennya diambil dari huruf ketiga dan keempat nomormahasiswa yang dituliskan yaitu **Ambiljur(Huruf34)**

Keenam store procedure ini kami simpan di server dan program yang kami buat dalam ASP tinggal memanggil saja. Dan mendapatkan hasilnya berupa data jadi misalnya begitu menjalankan cekpas maka jika hasilnya kosong berarti password tidak cocok.

Langkah yang perlu dilakukan di ASP untuk melakukan koneksi dengan database di server database dapat dilihat pada potongan kode dibawah ini:

```

strconn= "Driver={SQL
Server};SERVER=amikom32;UID=internet;PWD=AMIKOMYogyakarta;Database=amikom"
  
```

Pernyataan ini jika dieksekusi akan melakukan koneksi dari program kita ke database amikom di server database dengan nama amikom32, menggunakan user name "internet" dengan password "AMIKOMYogyakarta".

Setelah koneksi berhasil langkah selanjutnya adalah melakukan pengambilan data dengan cara mengeksekusi store procedure yang telah tersedia. Caranya dengan membuat object recordset baru yang secara detail dapat dilihat dibawah ini :

```

SQL1="Exec ambilnama '" & Request.Form{ltrim("NP")}
&"'"
  
```

```

set rsl=conn.execute(sql1)
  
```


Baris pertama menentukan dulu pernyataan SQLnya yang kalimatnya akan tergantung dari nomor mahasiswa yang dimasukkan pada form sebelumnya dengan nama variable NP, jika nomor mahasiswa yang diinputkan adalah 98.02.1200 maka kalimat SQLnya berbunyi : *Exec ambilnama "98.02.1200"* baris selanjutnya adalah pernyataan untuk menciptakan object recordset yang bernama rs1. object recordset ini berisi recordset yang merupakan hasil pengambilan data dengan mengeksekusi kalimat SQL yang dibuat pada baris sebelumnya.

Setelah data didapatkan maka langkah berikutnya adalah memanfaatkan data tersebut bisa dijadikan syarat percabangan bisa juga ditampilkan. Contoh recordset yang dijadikan syarat percabangan adalah sebagai berikut:

```
if rs1.eof then
```

Sedangkan contoh untuk menampilkan isi recordset dalam bentuk table adalah sebagai berikut :

```
<table border="1">
do while not rs.eof
no=no+1
<tr>
</strong><td width="24" align="center"
height="23" bgcolor="#E5E5E5"><em><font
face="Verdana"
size="1"><=no></font></em></td>
<td width="95" align="center" height="23"
bgcolor="#E5E5E5"><font face="Verdana"
size="1"><=rs("kode")></font></td>
<td width="350" align="left" height="23"
bgcolor="#E5E5E5"><font face="Verdana"
size="1"><=rs("mkl")></font></td>
<td width="111" align="center" height="23"
bgcolor="#E5E5E5"><font face="Verdana"
size="1"><=rs("sks")></font></td>
<td width="89" align="center" height="23"
bgcolor="#E5E5E5"><font face="Verdana"
size="1"><=rs("nilaitb")></font></td>
</tr>
<?
rs.movenext
loop
```

Untuk user yang di cantumkan dalam aplikasi KHS On-line kami buat dengan nama internet dan hak *execute* terhadap keenam store procedure tadi. Jadi walaupun user dapat mengakses kode program (*.asp), yang kami buat dan mereka

dapat melakukan login ke server database, dengan login id(*user name*) tersebut beserta paswordnya mereka tak dapat melakukan apapun.

PENUTUP

Demikian paparan aplikasi yang sederhana yang di bangun dengan disain keamanan maksimal untuk implementasi di internet maupun intranet. Pengembangan selanjutnya masih sangat luas dengan menyediakan fasilitas pembayaran on-line atau pengisian KRS via Web sehingga mahasiswa tidak perlu datang secara fisik atau pembangunan Perpustakaan dengan order peminjaman on-line sampai dengan e-commerce murni seperti penjualan buku atau penjualan jasa yang lain. Permasalahan yang sampai saat ini masih perlu di pikirkan banyak pihak adalah bagaimana memberikan jaminan keamanan transaksi tersebut sehingga data yang dikomunikasikan tadi tidak di bajak di jalan jika data tersebut menyangkut sesuatu yang berharga seperti nomor rekening atau kartu kredit sehingga transaksi lewat internet menjadi betul betul aman. Teknik teknik kriptography yang ada saat ini sampai SSL yang paling baru pun bagi kalangan informatika tidak akan terlalu lama untuk dibobol. Namun untuk transaksi data data yang tidak terlalu penting seperti KRS, KHS atau data pendaftaran rasanya kita tidak perlu terlalu risau terhadap isu keamanan tersebut karena justru keamanan data di server database yang dimiliki oleh organisasi pemilik data itulah yang perlu di jaga karena walaupun informasi itu jatuh ke tangan yang tidak berhak tidak akan berarti apa apa. Salah satu implementasi yang penulis anggap aman adalah implementasi yang penulis paparkan diatas.

DAFTAR PUSTAKA

1. David Solomon, Ray rankins, *SQL Server 6,5 Unleashed*, SAMS Publishing,1997
2. Jim Malloney, *Distributed COM application development with Visual Basic 6.0*, Prentice Hall, 1999
3. Michael Halvonsen, *Step by step Microsoft Visual Basic 6.0*, Microsoft press, 1999
4. WWW.ASPNet.com