

SISTEM INFORMASI KEPOLSIAN ONLINE BERBASIS JSP MENGGUNAKAN DATABASE TERDISTRIBUSI

Gede Galih Gumilang, Idris Winarno, S.ST, M.Kom, Wiratmoko Yuwono, ST

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember

Kampus PENS-ITS Keputih Sukolilo Surabaya 60111

Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114

Abstrak

Pembuatan SKCK dan pencarian kendaraan yang hilang adalah sebagian dari tugas kepolisian. Untuk mempermudah dalam menjalankan kegiatan tersebut maka telah dibuat sebuah sistem informasi database yang dapat mendistribusikan data dari berbagai polsek yang ada di Surabaya. Data yang dimaksud adalah data kepolisian terutama data kriminal dan data laporan kehilangan kendaraan dari masyarakat. Data pelaku kriminal di masukkan dari polsek-polsek yang ada di Surabaya kemudian disebar ke semua polsek, sehingga apabila ada seseorang tercatat pernah melakukan kejahatan pada suatu wilayah tertentu, maka seseorang tersebut tidak dapat membuat SKCK (Surat Keterangan Catatan Kepolisian) di seluruh polsek di Surabaya. Begitu juga dengan data kehilangan kendaraan, polsek memasukkan data tersebut kemudian didistribusikan melalui sistem informasi ini. Sehingga apabila suatu saat salah satu polsek menemukan kendaraan yang hilang tersebut maka dapat segera diketahui detail dari korban dengan melihat data kehilangan yang didistribusikan tadi. Karena data semacam ini sangat rahasia dan penting, maka pada sistem informasi ini telah dilengkapi dengan keamanan jaringan berupa HTTPS dan VPN.

1. Pendahuluan

1.1 Latar Belakang

Di era yang semakin pesat teknologi seperti saat ini, pertukaran informasi berjalan sangat cepat. Setiap orang membutuhkan data dari orang lain yang dapat digunakan untuk menjalankan sistem.

Kepolisian Republik Indonesia adalah sebuah instansi pemerintah yang mengeluarkan keterangan bahwa seorang warga negara dikatakan bermasalah atau tidak dengan hukum dan kriminal atau yang biasa dikenal dengan istilah Surat Keterangan Catatan Kepolisian (SKCK).

Disamping sebagai tempat pembuatan Surat Keterangan Catatan Kepolisian (SKCK), kepolisian adalah tempat untuk melaporkan apabila seorang warga mengalami kehilangan kendaraan bermotor. Banyak kasus, seorang warga negara yang mengalami kehilangan kemudian melaporkan kepada kepolisian terdekat, seseorang tersebut akan dipanggil oleh pihak kepolisian apabila kepolisian berhasil menemukan kendaraan bermotor tersebut dari hasil penangkapan yang dilakukan oleh.

Dengan kedua permasalahan seperti diatas maka telah dibangun suatu sistem informasi yang

menyimpan data pelaku kejahatan maupun data laporan kehilangan dalam satu kota/kabupaten yang datanya diinputkan oleh masing-masing Polsek dalam kota/kabupaten tersebut dan terintegrasi, sehingga dapat diakses juga oleh Polsek-polsek yang ada di kota/kabupaten tersebut. Apabila terjadi laporan dari suatu Polsek maka Polsek yang lain akan mengetahuinya saat itu juga.

1.2 Rumusan Permasalahan

Berdasarkan uraian diatas, maka permasalahan yang timbul dalam pengerjaan proyek akhir ini adalah :

1. Bagaimana mendapatkan data-data terkait yang akurat dan lengkap sebagai bahan untuk pengerjaan proyek akhir in.
2. Bagaimana membangun aplikasi sistem informasi ini yang berbasis web untuk proses penyimpanan data kriminal kepolisian sehingga dapat memberikan informasi kepada polsek-polsek yang ada dibawah jajaran Polrestabes Surabaya.
3. Bagaimana membangun sistem keamanan pada aplikasi ini sehingga dapat terhubung dengan aman.

1.3 Batasan Masalah

Batasan masalah pada proyek akhir ini adalah :

1. Daerah yang menjadi objek adalah wilayah kepolisian di bawah jajaran Polrestabes Surabaya dengan studi kasus pembuatan SKCK dan informasi kehilangan kendaraan bermotor.
2. Metode database yang digunakan adalah terdistribusi.
3. Menggunakan dua DBMS yaitu Oracle dan MySql.
4. Sistem informasi ini menggunakan JSP serta software tool pendukung lain.
5. Menggunakan keamanan jaringan HTTPS dan OpenVPN.

1.4 Tujuan dan Sasaran

Tujuan dari Proyek Akhir ini adalah untuk membantu kepolisian khususnya di wilayah Surabaya dalam mencocokkan data kriminal dan meningkatkan pengembalian barang bukti kepada korban agar dapat bermanfaat bagi pelayanan kepolisian. Selain itu juga untuk membantu manajemen data kriminal supaya tertata dengan baik.

1.5 Kontribusi Proyek

Aplikasi ini nantinya diharapkan dapat dikembangkan untuk menyelesaikan permasalahan pendistribusian data pelaku kriminal dan data laporan kehilangan kendaraan di wilayah Surabaya.

2. Teori Penunjang

2.1 Database Terdistribusi

2.1.1 Pengenalan Database Terdistribusi

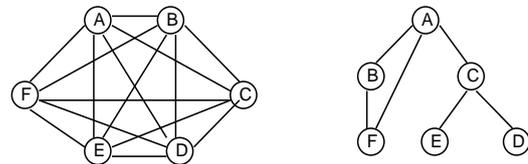
Pada basis data terdistribusi (*distributed database*), data disimpan pada beberapa tempat (*site*), setiap tempat diatur dengan suatu DBMS (*Database Management System*) yang dapat berjalan secara independent. Properti yang terutama terdapat pada basis data terdistribusi :

- Independensi data terdistribusi : pemakai tidak perlu mengetahui dimana data berada (merupakan pengembangan prinsip independensi data fisik dan logika).
- Transaksi terdistribusi yang atomic : pemakai dapat menulis transaksi yang mengakses dan mengubah data pada beberapa tempat seperti mengakses transaksi *local*.

Untuk tren basis data terdistribusi saat ini, pemakai harus mengetahui dimana data ditempatkan, juga harus mengetahui dimana sistem yang tidak mendukung independensi data terdistribusi dan transaksi terdistribusi atomik. Kedua *property* tersebut harus mendukung sistem secara efisien. Untuk sistem terdistribusi yang bersifat global, properti-properti tersebut kemungkinan tidak tepat karena adanya administrasi yang terlalu berlebihan dalam membuat lokasi data yang transparan.

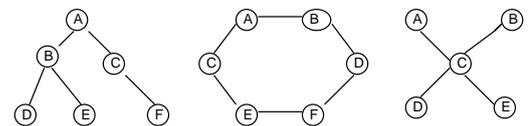
2.1.2 Model Arsitektur Jaringan

Site-site dalam database terdistribusi dihubungkan secara fisik dengan berbagai cara. Beberapa topologi digambarkan sebagai sebuah *graph* yang simpul-simpulnya bersesuaian dengan *site*. Sebuah *edge* dari simpul A ke simpul B bersesuaian dengan sebuah hubungan langsung antara dua *site*. Beberapa konfigurasi (bentuk) digambarkan sebagai berikut:



Fully Connected network

Partially connected network



Tree

Ring Network

Star Network

Gambar 2.1 Topologi network

Fully Connected network :

Keuntungan : kalau salah satu node rusak, yang lainnya masih dapat berjalan (tetapi biaya mahal).

Kerugian : *control* management tidak terjamin.

Partially connected network :

Keuntungan : reliability rendah, biaya dapat ditekan.

Kerugian : *control* management tidak terjamin.

Tree structure network :

Keuntungan : bersifat sentral, *control management* lebih terjamin

Kerugian : kalau node pusat (A) rusak, semua akan rusak.

Catatan : setiap proses dimulai dari bawah.

Ring Network (LAN) :

Keuntungan : rusak satu, yang lain masih berjalan.

Kerugian : *Control management* kurang terjamin karena bersifat desentralisasi.

Star Network (LAN) :

Keuntungan : - *control management* lebih terjamin, karena bersifat sentral.

- *reliability* rendah.

Kerugian : kalau pusat rusak, yang lainnya rusak.

2.1.3 Pentingnya Database Terdistribusi dan Kelebihannya.

Berikut ini alasan mengapa digunakan database terdistribusi:

1. Pengawasan distribusi dan pengambilan data.

Jika sejumlah *site* yang berbeda dihubungkan satu sama lain, lalu seorang pemakai yang berada pada satu *site* dapat mengakses data yang tersedia pada *site* lain.

Sebagai contoh : sistem distribusi pada sebuah bank memungkinkan seorang pemakai pada salah satu cabang dapat mengakses data cabang lain.

2. *Reliability* dan *availability*.

Sistem distribusi dapat terus menerus berfungsi dalam menghadapi kegagalan dari *site* individu atau mata rantai komunikasi antar *site*.

Misal : jika *site-site* gagal dalam sebuah sistem distribusi, *site-site* lainnya dapat melanjutkan operasi jika data telah direplikasi pada beberapa *site*.

3. Kecepatan pemrosesan *query*.

Jika sebuah *query* melibatkan data pada beberapa *site*, memungkinkan membagi *query* ke dalam *sub query* yang dapat dieksekusi dalam bentuk paralel oleh beberapa *site*. Perhitungan secara paralel mempercepat pemrosesan dari seorang pemakai *query*.

4. Otonomi lokal .

Pendistribusian sistem mengizinkan sekelompok individu dalam sebuah perusahaan untuk melatih pengawasan lokal melalui data mereka sendiri. Dengan kemampuan ini dapat mengurangi ketergantungan pada pusat pemrosesan.

5. Efisien dan fleksibel.

Data dalam sistem distribusi dapat disimpan dekat dengan titik di mana data tersebut dipergunakan. Data dapat secara dinamik bergerak atau disalin, atau salinannya dapat dihapus.

2.1.4 Tipe Database Terdistribusi

Terdapat dua tipe basis data terdistribusi :

- Homogen : yaitu sistem dimana setiap tempat menjalankan tipe DBMS yang sama
- Heterogen : yaitu sistem dimana setiap tempat yang berbeda menjalankan DBMS yang berbeda, baik Relational DBMS (RDBMS) atau non relational DBMS.

Gambaran basis data terdistribusi yang heterogen dapat dilihat pada Gambar 2.2.



Gambar 2.2 Database terdistribusi heterogen

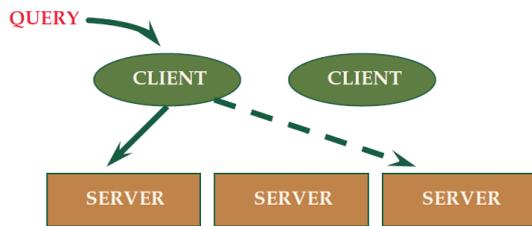
2.1.5 Arsitektur Database Terdistribusi

Terdapat tiga pendekatan alternatif untuk membagi fungsi pada proses DBMS yang berbeda. Dua Arsitektur alternatif DBMS terdistribusi adalah *Client/Server* dan *Collaboration Server*.

- *Client-Server*

Sistem *client-server* mempunyai satu atau lebih proses *client* dan satu atau lebih proses server, dan sebuah proses *client* dapat mengirim *query* ke sembarang proses server seperti pada Gambar 2.3. *Client* bertanggung jawab pada antar muka untuk user, sedangkan server mengatur data dan mengeksekusi transaksi. Sehingga suatu proses *client* berjalan pada sebuah personal computer dan

mengirim *query* ke sebuah server yang berjalan pada mainframe.



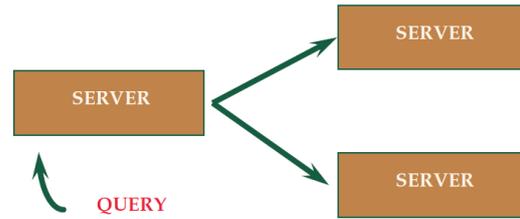
Gambar 2.3 Sistem *Client-Server*

Arsitektur ini menjadi sangat populer untuk beberapa alasan. Pertama, implementasi yang relatif sederhana karena pembagian fungsi yang baik dan arena *server* tersentralisasi. Kedua, mesin *server* yang mahal utilitasnya tidak terpengaruh pada interaksi pemakai, meskipun mesin *client* tidak mahal. Ketiga, pemakai dapat menjalankan antarmuka berbasis grafis sehingga pemakai lebih mudah dibandingkan antar muka pada *server* yang tidak *user-friendly*. Pada saat menulis aplikasi *client-server*, perlu diingat batasan antara *client* dan *server* dan untuk menjaga komunikasi antara keduanya yang berorientasi himpunan. Khususnya membuka kursor dan mengambil tupel pada satu waktu membangkitkan beberapa pesan dan dapat diabaikan.

- *Collaboration Server*

Arsitektur *client-server* tidak mengijinkan satu *query* mengakses banyak *server* karena proses *client* harus dapat membagi sebuah *query* ke dalam beberapa *subquery* untuk dieksekusi pada tempat yang berbeda dan kemudian membagi jawaban ke *subquery*. Proses *client* cukup kompleks dan terjadi *overlap* dengan *server*; sehingga perbedaan antara *client* dan *server* menjadi jelas. Untuk mengurangi perbedaan digunakan alternatif Arsitektur *client-server* yaitu sistem *Collaboration Server*. Pada sistem ini terdapat sekumpulan *server* basis data, yang menjalankan transaksi data lokal yang bekerjasama mengeksekusi transaksi pada beberapa server seperti pada Gambar 2.4.

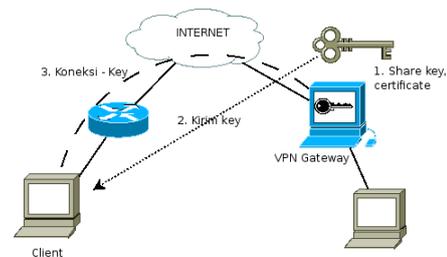
Jika *server* menerima *query* yang membutuhkan akses ke data pada server lain, sistem membangkitkan *subquery* yang dieksekusi *server* lain dan mengambil hasilnya bersama-sama untuk menggabungkan jawaban menjadi *query* asal.



Gambar 2.4 Sistem *Collaboration*

2.2 OpenVPN

OpenVPN adalah aplikasi open source untuk *Virtual Private Networking* (VPN), dimana aplikasi tersebut dapat membuat koneksi point-to-point tunnel yang telah terenkripsi. OpenVPN menggunakan *private keys*, *certificate*, atau *username/password* untuk melakukan autentikasi dalam membangun koneksi. Dimana untuk enkripsi menggunakan OpenSSL.



Gambar 2.5 Langkah-langkah VPN

Langkah-langkah membangun jaringan VPN adalah :

1. Pada VPN gateway membuat *shared key* dan *certificate*
2. Mengirimkan *key* tersebut kepada *client* yang akan melakukan koneksi
3. Membangun koneksi dengan menggunakan *key* yang telah didapat dari suatu VPN Gateway

2.3 HTTPS

Https adalah versi aman dari **HTTP**, protokol komunikasi dari World Wide Web.

Ditemukan oleh Netscape Communications Corporation untuk menyediakan autentikasi dan komunikasi tersandi dan penggunaan dalam komersi elektris. Selain menggunakan komunikasi *plain text*, HTTPS menyandikan data sesi menggunakan protokol SSL (*Secure Socket layer*) atau protokol TLS (*Transport Layer Security*). Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Pada umumnya port HTTPS adalah 443.

Tingkat keamanan tergantung pada ketepatan dalam mengimplementasikan pada *browser* web dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual. Oleh karena itu, pada halaman web digunakan HTTPS, dan URL yang digunakan dimulai dengan 'https://' bukan dengan 'http://'

Kesalahpahaman yang sering terjadi pada pengguna kartu kredit di web ialah dengan menganggap HTTPS "seungguhnya" melindungi transaksi mereka. Sedangkan pada kenyataannya, HTTPS hanya melakukan enkripsi informasi dari kartu mereka antara *browser* mereka dengan *web server* yang menerima informasi. Pada *web server*, informasi kartu mereka secara tipikal tersimpan di database *server* (kadang-kadang tidak langsung dikirimkan ke pemroses kartu kredit), dan server database inilah yang paling sering menjadi sasaran penyerangan oleh pihak-pihak yang tidak berkepentingan.

3. Rancangan Sistem

Dalam bab ini dibahas mengenai perencanaan dan pembuatan sistem informasi yang berhubungan dengan pendistribusian data dari satu DBMS ke DBMS yang lain serta JSP sebagai bahasa pemrograman berbasis web.

Didalam merancang sistem informasi tersebut, data pada database akan dirubah ke dalam file xml kemudian dikirim ke database lain. Sedangkan pembuatan sistem informasi itu sendiri dibangun menggunakan

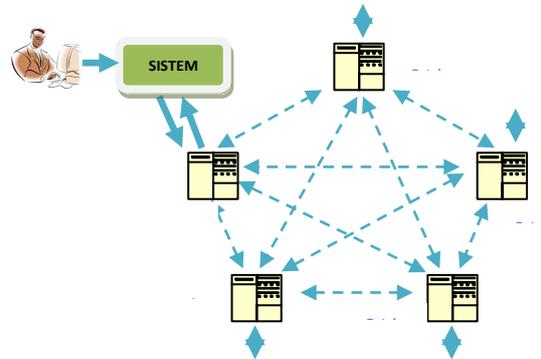
Pada tahap ini dijelaskan perancangan sistem tentang pendistribusian data antar DBMS yang nantinya dapat digunakan untuk pertukaran data serta sistem topologi yang akan menghubungkan seluruh DBMS yang ada di balik setiap sistem informasi.

Data yang dimasukkan ke dalam salah satu database akan dirubah kedalam sebuah file xml, kemudian file xml tersebut akan dikirim ke database lain. Saya menggunakan tipe file xml karena dengan file xml ini dapat dikirim ke DBMS yang berbeda. DBMS yang berbeda yang saya maksud adalah MySQL dan ORACLE. Sehingga file xml hasil konversi dari database MySQL akan dapat digunakan pada database ORACLE, begitu juga sebaliknya.

Keamanan pada sistem informasi ini terbagi ke dalam 3 tahap. Yang pertama adalah keamanan dari sistem informasi itu sendiri pada menu *login*. Yang kedua, karena sistem informasi ini berbasis *web* maka akan dibuat https. Keamanan yang

ketiga yaitu pada jaringan yang digunakan sistem informasi ini yaitu OpenVPN.

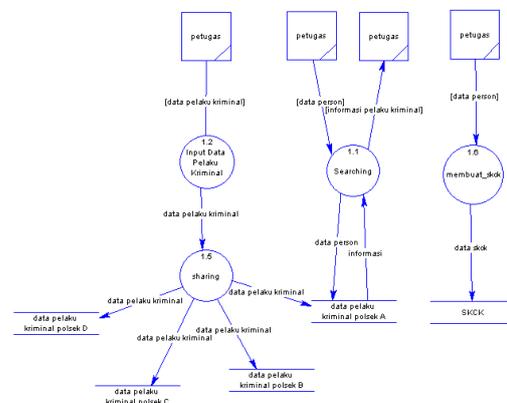
Penjelasan database terdistribusi secara umum melalui Gambar 3.1 :



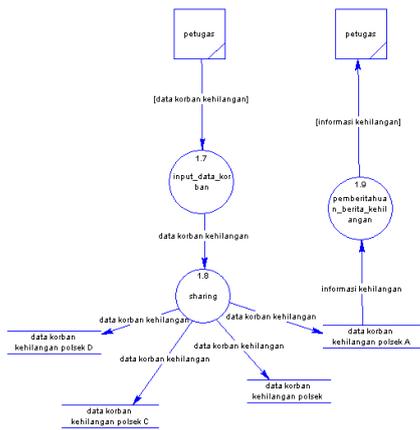
Gambar 3.1 Gambaran Sistem

Pada Gambar 3.1 dijelaskan bahwa setiap polsek di seluruh wilayah Surabaya terhubung melalui internet. Sehingga apabila ada update data dari salah satu polsek, maka data baru tersebut akan diketahui oleh seluruh polsek yang terhubung dengan sistem.

1. Petugas masuk ke dalam sistem informasi, kemudian melakukan *input* data atau mencocokkan data.
2. Sistem memasukkan data ke dalam database.
3. Sistem mengambil data dari database
4. Proses *sharing* data (metode replikasi gambar ditunjukkan gambar panah).
5. Data dari dan menuju sistem informasi pada masing-masing Polsek.

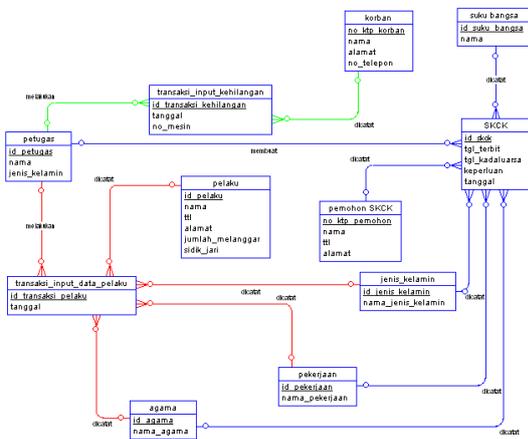


Gambar 3.2 DFD Level 1 proses *input* data pelaku kriminal, *searching* data dan pembuatan SKCK



Gambar 3.3 DFD Level 1 proses *input* data korban kehilangan.

Conceptual Data Model (CDM)



Gambar 3.4 Conceptual Data Model (CDM)

Gambar 3.4 merupakan conceptual data model dari sistem informasi ini. Terdapat 3 tabel transaksi yaitu *transaksi_input_kehilangan*, *transaksi_input_data_pelaku* dan pembuatan SKCK. Sedangkan tabel yang digunakan antara lain, *korban*, *petugas*, *petaku*, *jenis_kelamin*, *pekerjaan*, *agama*, *pemohon SKCK*, dan *suku bangsa*.

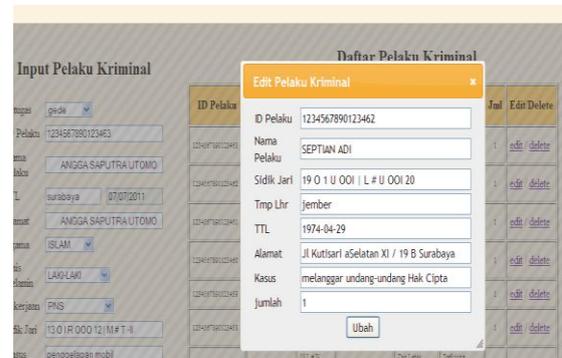
4. Pembuatan dan Analisa Sistem

Form *input* data pelaku kriminal



Gambar 4.1 Form *input* data pelaku kriminal

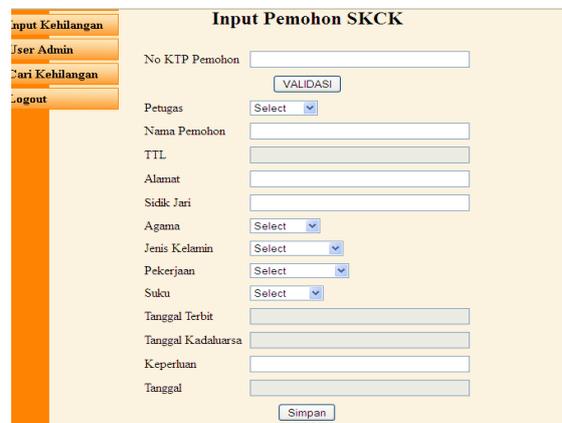
Input Pelaku, yaitu *form* yang digunakan untuk memasukkan data pelaku kriminal, selain itu juga dapat menghapus dan mengubah data pelaku kriminal yang ada seperti tampak pada Gambar 4.2.



Gambar 4.2 Kotak dialog edit

Pada form *input* data pelaku kriminal, petugas memasukkan nomor KTP pelaku, nama pelaku, TTL, alamat, agama, jenis kelamin, pekerjaan, kasus, pasal, rumus sidik jari, dan jumlah melanggar.

Form pembuatan SKCK.



Gambar 4.3 Form pembuatan SKCK

Gambar 4.3 merupakan form pembuatan SKCK. Pada form ini petugas memasukkan nomor KTP pemohon, kemudian petugas melakukan validasi dengan menekan tombol validasi. Apa bila nomor KTP yang dimasukkan tercatat pernah dimasukkan kedalam daftar pelaku criminal maka tombol simpan pada bagian bawah form akan hilang. Secara otomatis maka pemohon tidak dapat melakukan pembuatan SKCK.

Pada form ini petugas memasukkan nomor KTP pemohon, data nama pemohon, TTL, suku bangsa, agama, pekerjaan, alamat, nomor KTP, rumus sidik jari, keperluan, tanggal terbit, tanggal kadaluarsa. Pada saat petugas menekan tombol Simpan maka data akan tersimpan dan SKCK siap untuk di cetak.

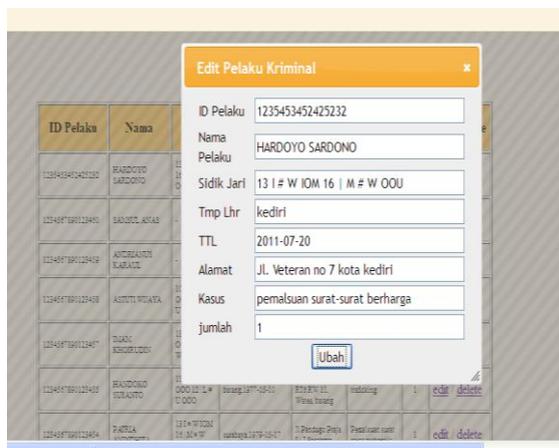
Proses update data

Pada proses insert, data awal adalah sebagai berikut :

ID Pelaku : 1235453452425232
 Nama : HARDOYO HDOKO
 Sidik Jari : 13 I # W IOM 16 | M # W OOU
 TTL : kediri,2011-07-20
 Alamat : Jl. Veteran no 7 kota kediri
 Kasus : pemalsuan surat-surat berharga
 Jml : 1

Kemudian akan merubah data sebagai berikut (perubahan nama) seperti yang terlihat pada gambar 4.14:

ID Pelaku : 1235453452425232
 Nama : HARDOYO SARDONO
 Sidik Jari : 13 I # W IOM 16 | M # W OOU
 TTL : kediri,2011-07-20
 Alamat : Jl. Veteran no 7 kota kediri
 Kasus : pemalsuan surat-surat berharga
 Jml : 1



Gambar 4.4 kotak dialog edit

Xml yang berubah :

```

-<data>
-<tabdata>
<ID_PELAKU>1235453452425232</ID_PELAKU>
<NAMA>HARDOYO HDOKO</NAMA>
<SIDIK_JARI>13 I # W IOM 16 | M # W OOU</SIDIK_JARI>
<TEMPAT_LAHIR>kediri</TEMPAT_LAHIR>
<TTL>2011-07-20</TTL>
<ALAMAT>Jl. Veteran no 7 kota kediri</ALAMAT>
<KASUS>pemalsuan surat-surat berharga</KASUS>
<JUMLAH_MELANGGAR>1</JUMLAH_MELANGGAR>
</tabdata>
</data>

```

Menjadi :

```

-<data>
-<tabdata>
<ID_PELAKU>1235453452425232</ID_PELAKU>
<NAMA>HARDOYO SARDONO</NAMA>
<SIDIK_JARI>13 I # W IOM 16 | M # W OOU</SIDIK_JARI>
<TEMPAT_LAHIR>null</TEMPAT_LAHIR>
<TTL>2011-07-20</TTL>
<ALAMAT>Jl. Veteran no 7 kota kediri</ALAMAT>
<KASUS>null</KASUS>
<JUMLAH_MELANGGAR>1</JUMLAH_MELANGGAR>
</tabdata>
</data>

```

4.1 Analisa Pengiriman Data.

4.1.1 Analisa proses insert.

Pada sistem informasi ini, data yang dikirim dirubah terlebih dahulu kedalam bentuk xml dengan format yang sederhana. Pada saat data disimpan, sistem informasi memasukkan data kedalam database pada sistem informasi itu sendiri, tetapi saat itu juga sistem menjalankan perintah untuk membuat suatu file xml dengan parameter inputan terakhir dengan menggunakan query sql sebagai berikut :

```

Select * from pelaku where modified_date =
(select max(modified_date)

```

Penjelasan query :

1. (select *) adalah perintah untuk mengambil semua nilai.
2. (pelaku) adalah nama tabel.
3. (where) adalah klausa untuk memasukkan parameter.
4. (modified_date) adalah field yang berisi nilai waktu. Nilai ini aku terisi dengan nilai sekarang (now()) pada saat petugas melakukan perintah insert.
5. (max(modified_date)) adalah perintah untuk menampilkan nilai waktu yang terbaru.

4.1.2 Analisa proses update.

Pada saat proses update, sistem melakukan pengecekan berdasarkan ID_PELAKU. Apabila saat petugas melakukan edit, maka sistem akan membuat sebuah persalinan data dari data yang lama ke data yang baru berdasarkan ID_PELAKU yang diproses. Setelah melakukan perubahan pada database yang terletak pada sistem informasi ini

sendiri, maka sistem akan melakukan perintah membuat file xml dari data baru tersebut. Sistem informasi lain akan selalu melihat apakah ada data baru yang dikirim. Apabila sistem informasi lain melihat ada data baru, maka sistem informasi lain tersebut mengambilnya untuk dilakukan proses parsing. Apabila ternyata parameter (ID_PELAKU) sudah ada pada database sebelumnya, maka akan dilakukan update data.

Berikut adalah sql query untuk mendapatkan data untuk diproses update :

```
Select * from pelaku where update_date =  
(select max(update_date) from pelaku)
```

Penjelasan query :

1. (select *) adalah perintah untuk mengambil semua nilai.
2. (pelaku) adalah nama tabel.
3. (where) adalah klausa untuk memasukkan parameter.
4. (update_date) adalah field yang berisi nilai waktu. Nilai ini akan terisi dengan nilai sekarang (now()) pada saat petugas melakukan perintah edit.
5. (max(update_date)) adalah perintah untuk menampilkan nilai waktu yang terbaru.

4.1.3 Analisa proses delete.

Proses delete sedikit lebih berbeda. Pada saat data diproses untuk dilakukan penghapusan, maka sistem hanya akan mengecek ID_PELAKU dan membuat file xml yang berisi nilai dari ID_PELAKU tersebut. Setelah berhasil jadi xml, maka sistem informasi lain selalu melihat dan memarsing data tersebut. Karena hanya berisi ID_PELAKU, maka sistem lain seperti mendapat perintah untuk melakukan proses hapus berdasarkan parameter ID_PELAKU tersebut.

Berikut adalah sql query untuk mendapatkan nilai sebuah parameter:

```
Select id_pelaku from pelaku where delete_date =  
(select max(delete_date) from pelaku)
```

Penjelasan query :

1. (select) adalah perintah untuk mengambil nilai.
2. (id_pelaku) adalah field dari tabel pelaku yang dijadikan parameter saat penghapusan data.
3. (pelaku) adalah nama tabel.
4. (where) adalah klausa untuk memasukkan parameter.

5. (delete_date) adalah field yang berisi nilai waktu. Nilai ini akan terisi dengan nilai sekarang (now()) pada saat petugas melakukan perintah edit.
6. (max(update_date)) adalah perintah untuk menampilkan nilai waktu yang terbaru.

5. Penutup

Pada bab-bab sebelumnya, mulai dari bab I sampai dengan bab IV telah diuraikan beberapa hal yang berhubungan dengan pembuatan sistem informasi ini, mulai dari latar belakang, dasar teori, perancangan dan pembuatan aplikasi, sampai dengan implementasinya yang disertai uji coba dan analisa. Pada bab ini diuraikan beberapa hal yang dapat disimpulkan dari hasil-hasil pengujian aplikasi dan beberapa saran dengan harapan untuk lebih menyempurnakan perancangan yang telah dibuat.

5.1 Kesimpulan

Berdasarkan hasil pengujian sistem yang telah dilakukan didapatkan beberapa kesimpulan pada sistem informasi ini.

1. XML dapat digunakan untuk pertukaran data multiplatform, sehingga dalam sistem informasi ini kami menggunakan XML sebagai sarana pertukaran data.
2. Pertukaran dengan menggunakan XML lebih ringan karena memiliki struktur yang lebih sederhana.
3. Penggunaan protocol HTTPS pada sistem informasi ini dapat membantu mengamankan pengiriman paket data antara client dan server.
4. Dengan menggunakan VPN membuat jaringan yang kita bangun menjadi aman karena berjalan pada jaringan privat.

DAFTAR PUSTAKA

- [1]. Ceri, Stefano & Pelagatti G, *Distributed Databases : Principles & Systems*, McGraw-Hill, Singapore, 1984
- [3]. Öszu, M.T & Valduriez, *Principles of Distributed Database Systems*, Prentice Hall, New Jersey, 1991
- [4]. lecturer.eepis-its.edu/~idris/HTTPS/SSL + (rev).
- [5]. lecturer.eepis-its.edu/~idris/OpenVPN + (rev).
- [6]. <http://openvpn.net/index.php/opensource/documentation/howto.html>