

## SENTRALISASI DATABASE DAFTAR HITAM: STUDI KASUS INTERNATIONAL REMITTANCE BANK

Moedjiono<sup>1</sup>, Anas Andriana Yusup<sup>2</sup>, Hendro Purwoko<sup>3</sup>

Magister Ilmu Komputer, Program pascasarjana, Universitas Budi Luhur  
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260  
Telp. (021) 5853753  
<sup>1</sup>moedjiono@gmail.com, <sup>2</sup>yusupanase@gmail.com, <sup>3</sup>hendroprwk08@gmail.com

### ABSTRAK

*PT. International Remittance Bank memiliki database-database untuk menyimpan semua daftar hitam baik nasional maupun internasional, permasalahannya adalah saat ini disimpan pada database-database yang terpisah yang menyebabkan pembuatan laporan semua daftar hitam menjadi lama, karena tidak adanya format data yang standar, koneksi database yang berbeda-beda, biaya investasi dan pemeliharaan yang tinggi untuk infrastruktur, network maupun sekuriti, sulit untuk melakukan monitoring, serta dilihat dari sisi arsitektur dan desain aplikasi dan database menjadi tidak efektif. Pada penelitian ini Penulis merancang database daftar hitam secara sentralisasi dengan menggunakan metode prototipe dimana database daftar hitam sentral akan menjadi pusat penyimpanan dari database-database daftar hitam yang ada pada PT. International Remittance Bank, hal ini sudah dilakukan pengujian menggunakan ISO 9126 dengan hasil prosentase 91,37% sangat baik untuk diimplementasikan. Perancangan database daftar hitam ini akan diimplementasikan pada tanggal 27 Maret 2014. Database daftar hitam sentralisasi akan memberikan manfaat diantaranya; menurunkan biaya investasi infrastruktur, jaringan maupun sekuriti; memudahkan dalam mengelola dan mengawasi database daftar hitam; arsitektur dan desain menjadi efektif; mudah dalam pengembangan aplikasi karena sudah ada standar format baik untuk pembuatan laporan maupun aplikasi pencegahan dini atau validasi yang terintegrasi terhadap transaksi yang mencurigakan (watch list filtering) dari semua daftar hitam yang ada; mengurangi resiko operasional, kredit dan resiko reputasi; untuk meningkatkan kepuasan kepada nasabah; serta mendukung pemerintah nasional maupun internasional dalam memberantas adanya transaksi yang mencurigakan baik Anti Money Laundering (AML), pendanaan teroris maupun kejahatan kriminal lainnya.*

**Kata Kunci :** Database, daftar hitam, watch list filtering, Anti Money Laundering (AML), ISO 9126

### 1. PENDAHULUAN

#### 1.1. Latar Belakang

Dalam bisnis perbankan tidak pernah lepas dari adanya sanksi terhadap nasabah-nasabah yang melakukan tindakan tertentu yang merugikan bagi pihak bank, mulai sanksi dari daftar hitam nasional (DHN), daftar hitam Bank Indonesia (DHBI), daftar hitam pencucian uang/Anti Money Laundering, daftar hitam jaringan teroris, daftar hitam OFAC dan lain sebagainya.

Indonesia masih masuk dalam daftar hitam negara-negara yang gagal memenuhi standar anti pencucian uang. Untuk keluar dari daftar tersebut, perumusan regulasi saja tidak cukup. Sistem keamanan jaringan komputasi sektor perbankan perlu segera ditingkatkan.

Head of Compliance for Actimize Europe, Trevor Barrit menjelaskan, posisi Indonesia yang masih masuk dalam daftar hitam badan pengawas pencucian uang, The Financial Action Task Force (FATF) karena belum sepenuhnya melakukan pengawasan terhadap regulasi yang berlaku.

Head of Compliance for Actimize Europe, Trevor Barrit menilai bank sentral dan lembaga keuangan di Indonesia juga harus patuh terhadap standar peraturan dunia. "Indonesia

belum memuaskan standar FATF, sebenarnya organisasi ini mengaharapkan Indonesia sudah bisa keluar dari daftar hitam ini," jelas di saat diskusi media bertajuk Anti Money Laundering & Fraud.

Berdasarkan FATF, Indonesia berada di urutan kesepuluh negara yang belum memenuhi standarisasi pencegahan pencucian uang pada tahun 2011. Angkanya naik signifikan. Pasalnya, pada tahun 2010, Indonesia masih menempati posisi 28. Bahkan, tercatat, per Juni 2013, Indonesia sudah berada di urutan kelima.

Iran menempati nomor satu, kemudian diikuti oleh Korea Utara, Ekuador, dan Ethiopia. "Sebenarnya FATF cukup mengapresiasi Indonesia karena telah menerbitkan undang-undang anti pencucian uang. Tapi masih ada masalah-masalah yang belum bisa diselesaikan," terang dia.

Adapun menurut laporan riset lembaga investigasi korporasi global, Kroll Inc, pada tahun 2012, 65% perusahaan Indonesia pernah mengalami penipuan di dunia maya (fraud) sekali dalam satu tahun. Perusahaan yang terkena fraud ini paling sering mengalami pencurian informasi internal .

Namun, langkah ini ternyata tidak cukup. Kunci penanganan fraud pada jaringan komputasi adalah selalu

mengedukasi sumberdaya manusia dan mengikuti perkembangan teknologi keamanan.

Begitu juga dengan International Remittance Bank yang merupakan salah satu bank asing terbesar di Indonesia dituntut untuk menerapkan pengelolaan terhadap resiko-resiko operasional dan kredit dari transaksi-transaksi yang mencurigakan baik dari dalam negeri maupun luarnegeri agar bisa memenuhi terhadap ketentuan dari regulator serta membuat nasabah merasa aman dan dalam melakukan transaksi perbankannya sehingga akan memberikan kepuasan yang terbaik terhadap para

## 2. LANDASAN TEORI DAN KERANGKA KONSEP

### 2.1. Tinjauan Pustaka

#### 2.1.1. Prototyping dan pemodelan perangkat lunak

Dalam proses analisis proses pendekatan dilakukan, dan salah satunya adalah *prototyping*. *Prototyping* adalah sebuah proses pengumpulan persyaratan, pengaplikasian prinsip analisis, dan penyusunan model perangkat lunak yang akan dibangun untuk penilaian dan pengembangan[1]. Akhirnya ada lingkungan yang membutuhkan konstruksi prototipe pada awal analisis, karena model adalah satu-satunya alat dimana persyaratan dapat ditarik secara efektif. Model tersebut kemudian dikembangkan dalam perangkat lunak produksi.

Ada empat model prototype:

- Prototype kertas, menggambarkan system dengan menggunakan media kertas. Prototype kertas tidak bisa diuji coba dan diimplementasikan.
- Prototype berbasis PC, memanfaatkan program aplikasi untuk menunjukkan interaksi manusia dan komputer.
- Prototype kerja, merupakan implementasi sebagian fungsi system yang ingin dilihat unjuk kerjanya, dan diwujudkan dalam sebuah program.
- Prototype program, program benar-benar dibuat dan dapat berfungsi dengan baik. Selain itu, program juga terus menerus ditambah dan dilengkapi.

Paradigma *prototyping* dapat terbatas atau tidak terbatas. Pendekatan terbatas disebut *throwaway prototyping*. Dengan menggunakan pendekatan ini, prototipe sebagai sebuah demonstrasi dari sebuah persyaratan. Sedangkan pendekatan tidak terbatas, yang disebut juga *evolutionary prototyping*., menggunakan prototipe sebagai bagian pertama dari aktivitas analisis yang akan diteruskan ke dalam desain dan konstruksi. Sebelum pendekatan terbatas atau tidak terbatas dilaksanakan perlu dilakukan apakah sistem yang akan dibangun dapat menerima prototyping atau tidak. Sejumlah faktor perlu diperhatikan, diantaranya: area aplikasi, kompleksitas aplikasi, karakteristik pelanggan, dan karakteristik proyek.

Beberapa kelebihan/manfaat yang bisa diambil bila kita menggunakan prototyping antara lain :

- Adanya komunikasi yang intensif antara pengembang dan user
- Membantu dalam analisis, karena kebutuhan user telah dipahami dengan baik oleh pengembang sehingga dapat meminimalkan salah persepsi antara kedua pihak.
- Peran user meningkat, karena user dapat melakukan evaluasi dan masukan baru setiap saat.

- Pengembangan lebih cepat, karena program bisa langsung dibuat dan user dapat melihat setiap tahap pembuatan program.
- Mudah dalam implementasinya, karena user sudah sejak awal terlibat sehingga sudah akrab dan tidak merasa asing terhadap program.

Sedangkan kelemahan memakai prototype adalah :

- User sibuk, karena user dan pengembang harus sama-sama memiliki komitmen untuk sering bertemu dan membahas kebutuhannya.
- User ingin program segera selesai sehingga pengembang sering mengabaikan dokumentasi.
- User berharap terlalu banyak, seringnya evaluasi dan komunikasi membuat user sering berubah pikiran dan tidak pasti akan kebutuhannya.

#### 2.1.2. Jaminan Kualitas Perangkat Lunak

Jaminan kualitas perangkat lunak (*Software Quality Assurance / SQA*) adalah aktivitas pelindung yang diaplikasikan pada seluruh proses perangkat lunak[2].

a. SQA meliputi :

- Pendekatan manajemen kualitas.
- Teknologi rekayasa perangkat lunak yang efektif (metode dan piranti).
- Kajian teknik formal yang diaplikasikan pada keseluruhan proses perangkat lunak.
- Strategi pengujian *multitiered* (deret bertingkat).
- Kontrol dokumentasi perangkat lunak dan perubahan yang dibuat untuknya.
- Prosedur untuk menjamin kesesuaian dengan standar pengembangan perangkat lunak (bila dapat diaplikasikan).
- Mekanisme pengukuran dan pelaporan.

Kualitas perangkat lunak didefinisikan sebagai konformansi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit, dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional.

Definisi tersebut berfungsi untuk menekankan tiga hal penting, yaitu :

- Kebutuhan perangkat lunak merupakan fondasi yang melaluinya **kualitas** diukur. Kurangnya penyesuaian terhadap kebutuhan juga menunjukkan rendahnya kualitas.
- Standar yang telah ditentukan menetapkan serangkaian kriteria pengembangan yang menuntun cara perangkat lunak direkayasa. Jika kriteria tersebut tidak diikuti, hampir pasti menimbulkan kualitas yang kurang baik.
- Ada serangkaian **kebutuhan implisit** yang sering tidak dicantumkan (misalnya kebutuhan akan kemampuan pemeliharaan yang baik). Bila perangkat lunak dapat berhasil menyesuaikan dengan kebutuhan eksplisitnya, tetapi gagal memenuhi kebutuhan implisitnya, maka kualitas perangkat lunak tersebut perlu diragukan.

b. Aktivitas SQA

Jaminan kualitas perangkat lunak terdiri dari berbagai tugas yang berhubungan dengan dua konstituen yang berbeda, perencana perangkat lunak yang mengerjakan kerja teknis dan kelompok SQA yang bertanggung jawab terhadap perencanaan jaminan kualitas, kesalahan, penyimpanan rekaman, analisis, dan pelaporan. Tugas kelompok SQA adalah membantu tim rekayasa perangkat lunak dalam pencapaian produk akhir yang berkualitas tinggi. The Software Engineering Institute merekomendasikan serangkaian aktivitas SQA yang menekankan rencana jaminan kualitas, kesalahan, penyimpanan rekaman, analisis, dan pelaporan.

Berikut ini aktivitas yang dilakukan (difasilitasi) oleh kelompok SQA yang independen :

1) Menyiapkan rencana SQA untuk suatu proyek.

Rencana itu dikembangkan selama perencanaan proyek dan dikaji oleh semua kelompok yang tertarik. Aktivitas jaminan kualitas yang dilakukan oleh tim rekayasa perangkat lunak dan kelompok SQA diatur oleh rencana.

Rencana tersebut mengidentifikasi hal-hal berikut :

- Evaluasi yang dilakukan
- Audit dan kajian yang dilakukan
- Standar yang dapat diaplikasikan pada proyek
- Prosedur untuk pelaporan dan penelusuran kesalahan
- Dokumen yang dihasilkan oleh kelompok SQA
- Jumlah umpan balik yang diberikan pada tim proyek perangkat lunak

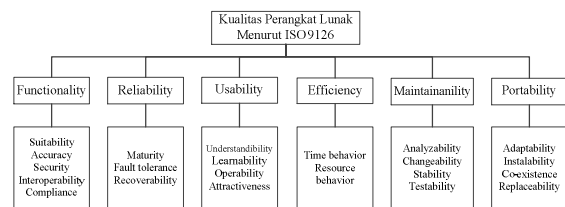
- 2) Berpartisipasi dalam pengembangan deskripsi proses pengembangan proyek. Tim rekayasa perangkat lunak memilih sebuah proses bagi kerja yang akan dilakukan.
- 3) Mengkaji aktivitas rekayasa perangkat lunak untuk memverifikasi pemenuhan proses perangkat lunak yang sudah ditentukan. Kelompok SQA mengidentifikasi, mendokumentasi, dan menelusuri deviasi proses dan membuktikan apakah koreksi sudah dilakukan.
- 4) Mengaudit produk kerja perangkat lunak yang ditentukan untuk membuktikan kesesuaian dengan produk kerja yang ditentukan tersebut sebagai bagian dari proses perangkat lunak.
- 5) Memastikan bahwa deviasi pada kerja dan produk kerja perangkat lunak didokumentasi dan ditangani sesuai prosedur pendokumentasian.
- 6) Mencatat ketidak-sesuaian dan melaporkannya kepada manajemen senior. Item-item yang tidak sesuai ditelusuri sampai item itu diubah.

2.1.3. ISO 9126

Kualitas perangkat lunak dapat dinilai melalui ukuran-ukuran dan metode-metode tertentu, serta melalui pengujian-pengujian software. Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu,

dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut [3] :

- a. *Functionality*(Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
- b. *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
- c. *Usability*(Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
- d. *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
- e. *Maintainability*(Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi.Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
- f. *Portability*(Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.



Gambar 1. Model Kualitas Perangkat Lunak Model ISO 9126

Masing-masing karakteristik kualitas perangkat lunak model ISO 9126 dibagi menjadi beberapa sub-karakteristik kualitas, yaitu:

Tabel 1. Karakteristik Kualitas Perangkat Lunak Model ISO 9126

Karakteristik	Sub-karakteristik	Deskripsi
<i>Functionality</i>	<i>Suitability</i>	Kemampuan perangkat lunak untuk menyediakan serangkaian fungsi yang sesuai untuk tugas-tugas tertentu dan tujuan pengguna.
	<i>Accuracy</i>	Kemampuan perangkat lunak dalam memberikan hasil

		yang presisi dan benar sesuai dengan kebutuhan.
	<i>Security</i>	Kemampuan perangkat lunak untuk mencegah akses yang tidak diinginkan, menghadapi penyusup ( <i>hacker</i> ) maupun otorisasi dalam modifikasi data.
	<i>Interoperability</i>	Kemampuan perangkat lunak untuk berinteraksi dengan satu atau lebih sistem tertentu.
	<i>Compliance</i>	Kemampuan perangkat lunak dalam memenuhi standar dan kebutuhan sesuai peraturan yang berlaku.
<i>Reliability</i>	<i>Maturity</i>	Kemampuan perangkat lunak untuk menghindari kegagalan sebagai akibat dari kesalahan dalam perangkat lunak.
	<i>Fault tolerance</i>	Kemampuan perangkat lunak untuk mempertahankan kinerjanya jika terjadi kesalahan perangkat lunak
	<i>Recoverability</i>	Kemampuan perangkat lunak untuk membangun kembali tingkat kinerja ketika terjadi kegagalan sistem, termasuk data dan koneksi jaringan.
<i>Usability</i>	<i>Understandibility</i>	Kemampuan perangkat lunak dalam kemudahan untuk dipahami.
	<i>Learnability</i>	Kemampuan

		perangkat lunak dalam kemudahan untuk dipelajari.
	<i>Operability</i>	Kemampuan perangkat lunak dalam kemudahan untuk dioperasikan.
	<i>Attractiveness</i>	Kemampuan perangkat lunak dalam menarik pengguna.
<i>Efficiency</i>	<i>Time behavior</i>	Kemampuan perangkat lunak dalam memberikan respon dan waktu pengolahan yang sesuai saat melakukan fungsinya.
	<i>Resource behavior</i>	Kemampuan perangkat lunak dalam menggunakan sumber daya yang dimilikinya ketika melakukan fungsi yang ditentukan.
<i>Maintainability</i>	<i>Analyzability</i>	Kemampuan perangkat lunak dalam mendiagnosis kekurangan atau penyebab kegagalan.
	<i>Changeability</i>	Kemampuan perangkat lunak untuk dimodifikasi tertentu.
	<i>Stability</i>	Kemampuan perangkat lunak untuk meminimalkan efek tak terduga dari modifikasi perangkat lunak.
	<i>Testability</i>	Kemampuan perangkat lunak untuk dimodifikasi dan divalidasi perangkat lunak lain.
<i>Portability</i>	<i>Adaptability</i>	Kemampuan perangkat lunak



		untuk diadaptasikan pada lingkungan yang berbeda-beda.
	<i>Instalability</i>	Kemampuan perangkat lunak untuk diinstal dalam lingkungan yang berbeda-beda.
	<i>Coexistence</i>	Kemampuan perangkat lunak untuk berdampingan dengan perangkat lunak lainnya dalam satu lingkungan dengan berbagi sumber daya.
	<i>Replaceability</i>	Kemampuan perangkat lunak untuk digunakan sebagai pengganti perangkat lunak lainnya.

ISO 9126 adalah standar terhadap kualitas perangkat lunak yang diakui secara internasional. Terpenuhinya item-item pada ISO 9126 pada sebuah perangkat lunak tidak serta merta memberikan sertifikat ISO terhadap perangkat lunak tersebut karena standar ISO juga harus dipenuhi dari sisi manajemen pembuat perangkat lunak tersebut, dengan kata lain jika manajemennya tidak memenuhi standar ISO maka hasil kerjanya pun tidak dapat diberikan sertifikat standar ISO.

Faktor-faktor ISO 9126 tidak serta merta memungkinkan kita untuk melakukan pengukuran kualitas secara langsung. Meskipun demikian, standar tersebut menyediakan basis yang sangat penting untuk melakukan pengukuran-pengukuran kualitas secara tidak langsung dan pada dasarnya menyediakan daftar yang sempurna untuk menilai kualitas suatu sistem/perangkat lunak.

Pengujian kualitas sistem dilakukan untuk menguji tingkat kualitas perangkat lunak sistem informasi yang dihasilkan berdasarkan empat karakteristik kualitas perangkat lunak yang terdapat pada ISO 9126, yaitu *functionality*, *reliability*, *usability*, dan *efficiency*.

Konsep ISO 9126 merupakan konsep terbaru dibandingkan konsep-konsep lainnya, sehingga konsep ini lebih relevan dengan situasi perkembangan teknologi informasi saat ini. konsep ISO 9126 sudah menjadi standar internasional dalam mengevaluasi kualitas perangkat lunak.

Hasil identifikasi ISO 9126, dari enam karakteristik kualitas sebuah aplikasi ditetapkan hanya empat karakteristik saja yang dijadikan variabel dalam penelitian ini, yaitu

*functionality*, *reliability*, *usability*, dan *efficiency*. Dua karakteristik lainnya yaitu *maintainability* dan *portability* tidak menjadi fokus penelitian. Pengujian hanya dilakukan pada penggunaan aplikasi sistem informasi manajemen kepegawaian dari sisi *client* dan tidak masuk dalam lingkup *server*. Pengujian kualitas perangkat lunak dilakukan untuk menguji hipotesis kedua dalam penelitian ini.

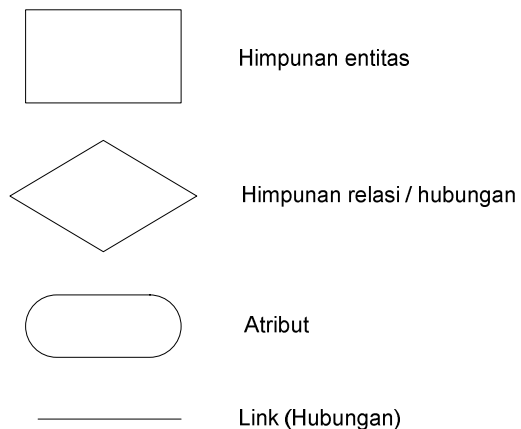
**2.1.4. Database Management System**

Basis data (database) adalah kumpulan dari data-data yang saling berhubungan satu dengan yang lainnya, dan tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya [4]. Database adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu[4].

Pada perancangan basis data, model yang dilakukan dengan menggunakan model konseptual secara model data relational. Terdapat dua buah teknik perancangan, yaitu :

**a) Teknik Entity Relationship / Teknik Perancangan Basis Data Pendekatan E-R**

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan data dan hubungan antar data. ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antar data. Pada dasarnya ada tiga macam simbol yang digunakan, yaitu :



Gambar 2. Simbol Diagram ER

**b) Teknik Normalisasi**

Proses normalisasi adalah proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Pada proses normalisasi dilakukan pengujian pada beberapa kondisi apakah ada kesulitan pada saat menambah/menyisipkan, menghapus, mengubah dan mengakses pada suatu basis data. Bila terdapat kesulitan pada pengujian tersebut maka perlu dipecahkan relasi pada beberapa tabel lagi atau dengan kata lain perancangan basis data belum optimal.

Tujuan dari normalisasi itu sendiri adalah untuk menghilangkan kerangkapan data, mengurangi kompleksitas, dan untuk mempermudah pemodifikasian data.

### 2.1.5. Konsep Dasar Basis Data Terpusat (*Centralized Database*)



Gambar 3. Arsitektur Basis Data Terpusat [5]

Sistem basis data terpusat merupakan suatu sistem yang menempatkan data di suatu lokasi saja dan semua lokasi lain mengakses basis data di lokasi tersebut. Dimana tugas *server* disini hanya melayani permintaan dari *client*. Sedangkan *client* meminta layanan dari *server*.

Implementasi *Distributed data processing* saat ini telah lebih maju dari saat awal konsep ini diperkenalkan. Dengan menggunakan database *client/server*, Sebuah *server* berfungsi jauh lebih efisien tidak hanya sebatas *file server*, melainkan ikut pula berperan aktif dalam pengolahan data, dan mengelola hak akses *user* atas data. Hal ini menjadikan fungsi *server* sebagai *file server* berubah menjadi data *server*.

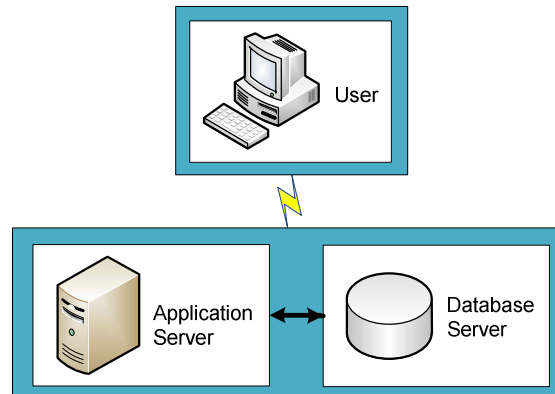
Namun perkembangan saat ini, banyak pihak memilih untuk lebih memaksimalkan fungsi server dalam pengolahan data. Dengan memanfaatkan secara penuh keberadaan *stored procedure*, *trigger*, dan *user defined functions*, *developer* dapat meningkatkan kemampuan *server* lebih dari sekedar melakukan eksekusi *query* atau pemrosesan data sederhana, melainkan menjadikannya sebagai *business logic* atas sistem yang dibangun yang dalam implementasinya lebih dikenal sebagai *centralized data processing*.

Implementasi *centralized dataprocessing* ini dipilih karena menawarkan beberapa keuntungan:

- a. Kinerja sistem yang lebih terjamin, tidak seperti pada *implementation distributed data procesing*, implementasi *centralized data procesing* tidak terpengaruh pada spesifikasi teknis hardware disisi *client* yang digunakan. Aplikasi disisi *client* hanya merupakan *thin client* yang berfungsi sebagai *input/output* terminal.
- b. Efisiensi dalam pemanfaatan *network traffic*, Pada implementasi *distributed data procesing*, fungsi *server* hanya mengolah data secara sederhana dan umumnya memberikan data mentah untuk kemudian diolah di sisi *client*. Hal ini menyebabkan *network traffic* menjadi penuh dan kerap kali menjadi *bottleneck* (Respons dari aplikasi menjadi lebih lambat) dan salah satu faktor penyebab terbesar turunnya kinerja sistem yang digunakan. Dengan mengimplementasikan *centralized data processing*, fungsi server hanya memberikan data jadi. *Client* hanya bertugas untuk mengolah data tersebut ke dalam media output seperti laporan atau

mentransfernya ke dalam format data lain yang dibutuhkan.

- c. Kemudahan dalam *maintenance* dan *deployment*. Dengan *implementasi centralized data processing*, fungsi-fungsi penting diletakkan di sisi server. Hal ini tentunya meningkatkan kemudahan dalam *maintenance* sistem yang dibangun karena tidak memerlukan *redployment* saat ada perubahan yang harus dilakukan pada sisi *business logic* yang digunakan. Karena perubahan tersebut hanya perlu dilakukan di sisi *server*.
- d. *Cost saving*, Implementasi *centralized data processing* tidak membutuhkan keberadaan mesin-mesin client dengan spesifikasi teknis tinggi.
- e. Menghapus redundansi dan menyederhanakan pemeliharaan system <http://log.web.id/cgi-bin/mt-tb.cgi/39>.



Gambar 4. Arsitektur Basis Data Terpusat[6]

Sebuah sistem basis data terpusat hanya mungkin dibangun dalam sebuah sistem jaringan komputer. Dalam sebuah sistem jaringan komputer kita mengenal adanya topologi, yang akan menentukan bagaimana konfigurasi/keterhubungan antara satu simpul jaringan (*node/site*) dengan simpul-simpul lainnya.

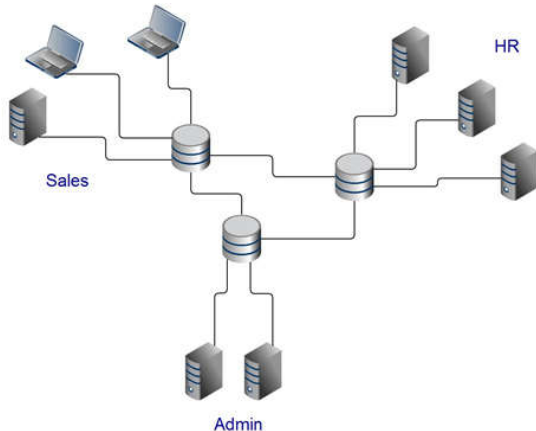
Sistem basis data terpusat merupakan suatu konsep *client/server* dengan model *two-tier* (dua tingkatan). Dua Tingkatan adalah proses dimana komputer *workstation* membuat hubungan ke komputer *server* dan mempertahankan hubungan tersebut sampai proses selesai. Misalnya pada proses pemasukan data atau membuat sebuah laporan. Model *two tier* terdiri atas sebuah server dan banyak *client* yang dihubungkan melalui jaringan. Aplikasi ditempatkan dikomputer *client* dan mesin database ditempatkan di server. *Tier client* disebut juga tingkatan presentasi. *Tier client* berisi kode yang akan menampilkan data dan interaksi dengan *user*. Tugas yang terjadi pada *client* adalah : antarmuka pengguna, interaksi *database*, pengambilan dan modifikasi data, aturan bisnis, dan pengaturan kesalahan.

Hubungan yang terbentuk sangat konsumtif dalam sumber daya (alokasi memori, ruang hardisk, kontrol, dan lainnya), karena tidak efektif jika dilakukan untuk hubungan

yang melibatkan banyak pemakai (contohnya pada sistem jaringan internet). Oleh karena itu tidak heran jika konsep dua Tingkatan atau (*two tier*) hanya dipraktekkan pada LAN (*Local Area Network*).

**2.1.6. Basis Data Terdistribusi (*Distributed DBMS*)**

Secara logik keterhubungan dari kumpulan-kumpulan data yang digunakan bersama-sama, dan didistribusikan melalui suatu jaringan komputer.

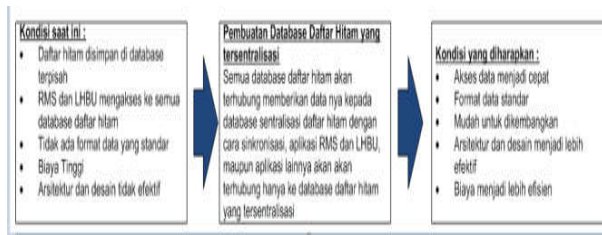


Gambar 5. Basis Data Terdistribusi[7]

Sebuah sistem perangkat lunak yang mengatur basis data terdistribusi dan membuat pendistribusian data secara transparan.DDBMS memiliki satu logikal basis data yang dibagi ke dalam beberapa fragment. Dimana setiap fragment disimpan pada satu atau lebih komputer dibawah kontrol dari DBMS yang terpisah , dengan mengkoneksi komputer menggunakan jaringan komunikasi.

**2.1.7. Kerangka Konsep**

Kerangka pemikiran penulisan tesis ini dapat dilihat pada gambar 6.



Gambar 6. Kerangka Konsep Pemikiran

Gambar 6 merupakan kerangka pemikiran yang dapat diuraikan sebagai berikut :

Usulan dalam penelitian ini adalah perlunya dibuatkan sebuah database daftar hitam yang tersentralisasi yang dapat menyimpan semua informasi daftar hitam yang ada pada International Remittance Bank, sehingga akan memberikan solusi pada permasalahan pada database daftar hitam saat ini.

Pembuatan database daftar hitam yang tersentralisasi ini diharapkan dapat menurunkan biaya investasi dan pemeliharaan, mudah dalam pemeliharaan dan pengawasan, karena aplikasi RMS dan LHBU hanya terkoneksi ke satu database saja, akses data menjadi lebih cepat, tersedianya format data yang standar jadi mempercepat *performance*, segi arsitektur dan desain menjadi lebih efektif dan efisien, serta mudah untuk dikembangkan, kedepan bisa saja akan bertambah daftar hitam baru seperti daftar hitam dari vendor pengadaan barang di Indonesia, daftar hitam dari kepolisian, daftar hitam kejahatan asian tenggara dan daftar hitam lainnya.

**3. METODOLOGI DAN RANCANGAN PENELITIAN**

**3.1. Metodologi Penelitian**

Penelitian ini menggunakan metode analisis deskriptif yang bertujuan untuk mendapatkan gambaran yang lebih mendalam dan lengkap dari obyek yang akan diteliti dengan melakukan pengamatan (*observasi*) secara langsung di lapangan, melakukan wawancara (*interview*) kepada bagian yang terkait langsung dan melakukan studi literatur.

**3.2. Sampling/Metode Pemilihan Sample**

Proses pemilihan sampling yang digunakan pada penelitian ini adalah menggunakan *Non-Random Sampling* secara *Convenience Sampling*, dimana Penulis yang menentukan narasumbernya yang merupakan *subject matter expert* dari IT Directory International Remittance Bank.

**3.3. Metode Pengumpulan Data**

Metode pengumpulan datanya adalah dengan mengumpulkan data secara primer dari para narasumber IT Directory International Remittance Bank pada saat wawancara dan dokumen-dokumen yang sudah diberikan kepada Penulis.

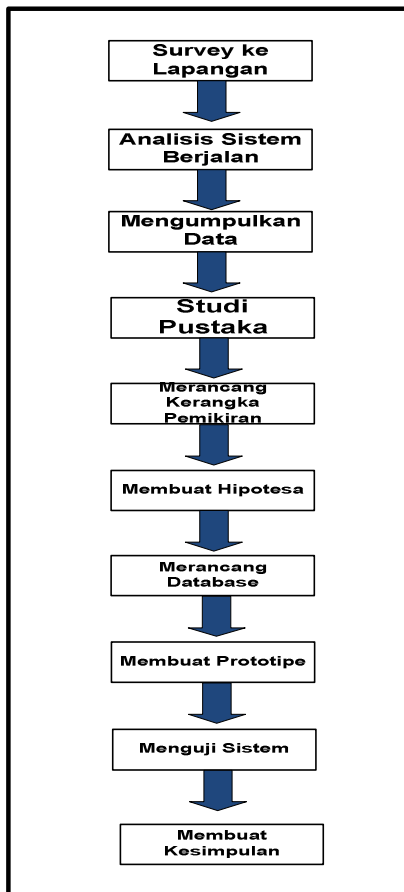
**3.4. Instrumentasi**

Data dan informasi penelitian ini diperoleh melalui studi lapangan yang bersifat primer yaitu melalui dengan melakukan wawancara kepada para narasumber.

**3.5. Teknis Analisis**

Teknik analisis yang digunakan adalah melakukan pengujian prototipe menggunakan ISO 9126 sebagai uji kualitasnya untuk menentukan layak atau tidak layak suatu perangkat lunak diimplementasikan..

**3.6. Langkah-langkah Penelitian**



Gambar 7. Langkah-langkah penelitian

Penjelasan dari deskripsi langkah penelitian tersebut adalah :

1. Melakukan survey ke lapangan  
Langkah ini bertujuan untuk mengetahui kondisi di lapangan pada International Remittance Bank. Survei awal ini untuk mengetahui permasalahan yang ada, merumuskan serta menentukan topik penelitian yang akan diambil oleh Penulis.
2. Melakukan analisis terhadap sistem yang berjalan  
Penelitian dimulai dengan melakukan pengamatan pada database daftar hitam yang sedang berjalan.
3. Mengumpulkan Data  
Mengumpulkan data dengan melakukan pengamatan langsung dan wawancara terhadap para narasumber yang merupakan *key person*.
4. Melakukan studi pustaka  
Melakukan studi pustaka yang berkaitan dengan pembuatan database sentralisasi. Studi pustaka dilakukan dengan mempelajari konsep database.
5. Merancang Konsep Pemikiran  
Merancang pembuatan database daftar hitam tersentralisasi sebagai usulan dari Penulis, yang diharapkan akan memberikan solusi yang lebih baik dibandingkan dengan database daftar hitam yang sedang berjalan.

6. Membuat Hipotesis  
Melakukan kegiatan hipotesis atas kerangka pemikiran yang telah dibuat untuk proposal tesis ini agar bisa dilanjutkan menjadi naskah akhir penulisan tesis.
7. Merancang Database  
Membuat rancangan database daftar hitam sesuai yang diusulkan Penulis.
8. Membuat Prototipe  
Membuat prototipe untuk pengujian dalam membandingkan kecepatan respon database antara database daftar hitam saat ini dengan rancangan database yang diusulkan Penulis.
9. Menguji Sistem  
Melakukan pengujian terhadap prototipe dengan menggunakan ISO 9126 sebagai uji kualitas untuk menentukan layak atau tidaknya perangkat lunak diimplementasikan.
10. Membuat Kesimpulan  
Tahapan terakhir adalah membuat kesimpulan dari penelitian ini.

#### 4. PEMBAHASAN HASIL PENELITIAN

##### 4.1. Analisis Sistem

Kebutuhan fungsional dan non-fungsional dalam perancangan sentralisasi database daftar hitam pada PT. International Remittance Bank adalah :

- a. Membuat Database Sentralisasi Daftar Hitam untuk menyimpan data daftar hitam DHN, DHBI, AML, dan OFAC.
- b. Impor database dari database DHN, DHBI, AML, dan OFAC ke database daftar hitam sentral.
- c. Membuat simulasi validasi daftar hitam yang dapat mengakses ke database daftar hitam central
- d. Menyediakan Server database untuk server utama dan server DR.
- e. Menyediakan Network dan Firewall Konfigurasi agar masing-masing database dapat saling berkomunikasi.
- f. Menyediakan storage untuk data archiving.

##### 4.2. Rancangan Sistem

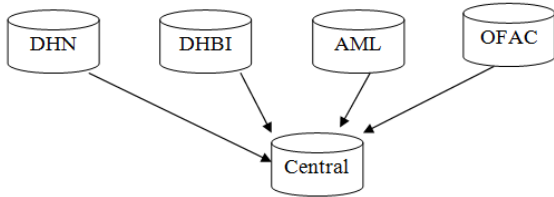
###### 4.2.1. Pengumpulan data dan analisa

Database daftar hitam saat ini terdiri dari :

- Database DHN disimpan di SQL Server 2005, digunakan untuk menyimpan data daftar hitam transaksi nasabah giro secara nasional.
- Database DHBI disimpan di SQL Server 2005, digunakan untuk menyimpan data daftar hitam kredit macet secara nasional.
- Database AML disimpan di SQL Server 2005, digunakan untuk menyimpan data daftar hitam transaksi pencucian uang secara nasional dan internasional.
- Database OFAC disimpan di Oracle 11g., digunakan untuk menyimpan data daftar hitam negara-negara yang di-embargo oleh Amerika Serikat.

###### 4.2.2. Perancangan database secara konseptual

Database daftar sentral merupakan penggabungan dari database daftar hitam DHN, DHBI, AML, dan OFAC.



Gambar 8. Perancangan database secara konseptual

**4.2.3. Pemilihan Database**

Database yang akan dipilih dalam perancangan sentralisasi database daftar hitam pada PT.International Remittance Bank adalah menggunakan SQL Server 2010, karena lebih mudah untuk bisa interkoneksi dengan database-database lain yang berbeda platform maupun operating system.

**4.2.4. Perancangan database secara logic**

Tabel 2. Atribut tabel DHN

No	Field	Data Type	Length	Description
1	DHN_ID	Char	10	Primary Key
2	DHN_ID_Type	Char	2	Primary Key
3	DHN_Name	Char	50	
4	DHN_DOB	Date	8	
5	DHN_Gender	Char	50	
6	DHN_Address	Char	50	
7	DHN_City	Char	50	
8	DHN_Province	Char	50	
9	DHN_Zipcode	Char	10	
10	DHN_Bank_Code	Char	10	
11	DHN_Segment_Type	Char	10	
12	DHN_Date	Date	8	
13	DHN_Bank_Account	Char	20	
14	DHN_Cheque_Number	Char	10	
15	DHN_Cheque_Amount	Char	20	
16	DHN_Verbal Id	Char	50	
17	DHN_Remark	Char	50	
18	DHN_Occupation	Char	50	
19	DHN_Remark	Char	50	

Tabel 3. Atribut tabel DHBI

No.	Field	Data Type	Length	Description
1	DHBI_ID	Char	10	Primary Key
2	DHBI_ID_Type	Char	2	Primary Key
3	DHBI_Name	Char	50	
4	DHBI_DOB	Date	8	

Tabel 4. Atribut tabel AML

No.	Field	Data Type	Length	Description
1	AML_ID	Char	10	Primary Key
2	AML_ID_Type	Char	2	Primary Key
3	AML_Country_ID	Char	3	Primary Key
4	AML_First_Name	Char	50	
5	AML_Middle_Name	Char	50	
6	AML_Last_Name	Char	50	
7	AML_DOB	Date	8	
8	AML_Verbal_Id	Char	50	
9	AML_Criminal	Char	50	
10	AML_Gender	Char	2	
11	AML_Segment_Type	Char	10	
12	AML_Occupation	Char	50	
13	AML_Religion	Char	10	
14	AML_Date	Date	8	
15	AML_Bank_Code	Char	50	
16	AML_Bank_Account	Char	20	
17	AML_Type_Blood	Char	2	
18	AML_Address	Char	50	
19	AML_Zipcode	Char	10	
20	AML_City	Char	50	
21	AML_Province	Char	50	
22	AML_Status	Char	50	
23	AML_Remark	Char	50	
24	AML_Swift_Code	Char	50	

Tabel 5. Atribut tabel OFAC

No.	Field	Data Type	Length	Description
1	OFAC_ID	Char	10	Primary Key
2	OFAC_ID_Type	Char	2	Primary

				Key
3	OFAC_Country_ID	Char	3	Primary Key
4	OFAC_First_Name	Char	50	
5	OFAC_Middle_Name	Char	50	
6	OFAC_Last_Name	Char	50	
7	OFAC_DOB	Date	8	
8	OFAC_Verbal_Id	Char	50	
9	OFAC_Gender	Char	2	
10	OFAC_Segment_Type	Char	2	
11	OFAC_Occupation	Char	50	
12	OFAC_Address	Char	50	
13	OFAC_City	Char	10	
14	OFAC_Date	Date	8	
15	OFAC_Bank_Code	Char	50	
16	OFAC_Bank_Account	Char	20	
17	OFAC_Province	Char	50	
18	OFAC_Status	Char	50	
19	OFAC_Zipcode	Char	10	
20	OFAC_Swift_Code	Char	50	
21	OFAC_Remark	Char	50	

Selanjutnya dari 4 struktur tabel yang ada, akan dilakukan proses normalisasi data yaitu proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Pada proses normalisasi dilakukan pengujian pada beberapa kondisi apakah ada kesulitan pada saat menambah/menyisipkan, menghapus, mengubah dan mengakses pada suatu basis data. Bila terdapat kesulitan pada pengujian tersebut maka perlu dipecahkan relasi pada beberapa tabel lagi atau dengan kata lain perancangan basis data belum optimal. Tujuan dari normalisasi itu sendiri adalah untuk menghilangkan kerangkapan data, mengurangi kompleksitas, dan untuk mempermudah pemodifikasian data.

Bentuk tidak normal dari perancangan database ini adalah :

AML_ID	OFAC_ID	DHBI_ID	DHN_ID
AML_ID_Type	OFAC_ID_Type	DHBI_ID_Type	DHN_ID_Type
AML_Count_ry_ID	OFAC_Count_ry_ID	DHBI_Name	DHN_Name
AML_First_Name	OFAC_First_Name	DHBI_DOB	DHN_DOB
AML_Middle_Name	OFAC_Middle_Name	DHBI_Gender	DHN_Gender
AML_Last_Name	OFAC_Last_Name	DHBI_Address	DHN_Address
AML_DOB	OFAC_DOB	DHBI_City	DHN_City
AML_Verbal_Id	OFAC_Verbal_Id	DHBI_Province	DHN_Province
AML_Criminal	OFAC_Gender	DHBI_Zipcode	DHN_Zipcode
AML_Gender	OFAC_Segment_Type	DHBI_Bank_Code	DHN_Bank_Code
AML_Segm	OFAC_Occu	DHBI_Segm	DHN_Segme

ent_Type	ation	ent_Type	nt_Type
AML_Occupation	OFAC_Address	DHBI_Date	DHN_Date
AML_Religion	OFAC_City	DHBI_Facility	DHN_Bank_Account
AML_Date	OFAC_Date	DHBI_Collectibility	DHN_Cheque_Number
AML_Bank_Code	OFAC_Bank_Code	DHBI_Status	DHN_Cheque_Amount
AML_Bank_Account	OFAC_Bank_Account	DHBI_Verbal Id	DHN_Verbal Id
AML_Type_Blood	OFAC_Province	DHBI_Currency	DHN_Remark
AML_Address	OFAC_Status	DHBI_Occupation	DHN_Occupation
AML_Zipcode	OFAC_Zipcode	DHN_Remark	DHN_Remark
AML_City	OFAC_Swift_Code	DHBI_Outstanding	
AML_Province	OFAC_Remark	DHBI_Status_Nikah	
AML_Status	OFAC_Married_Status		
AML_Remark			
AML_Swift_Code			
AML_Married_Status			

Bentuk Bentuk Normal Kesatu (1NF) mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa "atomic value". Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (multi value). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu artis saja dan juga bukanlah pecahan kata kata sehingga artinya lain. Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

Bentuk 1NF adalah :

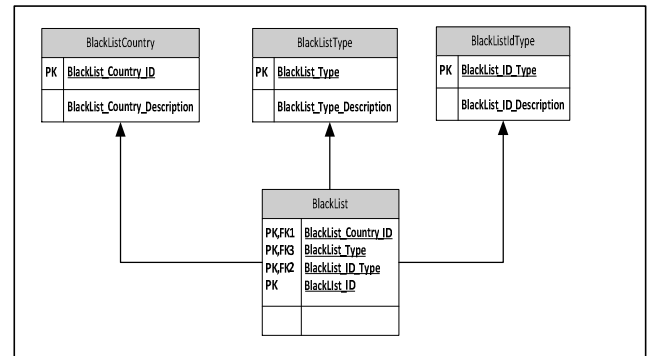
AML (AML\_ID, AML\_ID\_Type, AML\_Country\_ID, AML\_First\_Name, AML\_Middle\_Name, AML\_Last\_Name, AML\_DOB, AML\_Verbal\_Id, AML\_Criminal, AML\_Gender, AML\_Segment\_Type, AML\_Segment\_Type, AML\_Occupation, AML\_Religion, AML\_Date, AML\_Bank\_Code, AML\_Bank\_Account, AML\_Type\_Blood, AML\_Address, AML\_Zipcode, AML\_City, AML\_Province, AML\_Status, AML\_Status, AML\_Remark, AML\_Remark, AML\_Swift\_Code, AML\_Married\_Status)  
 OFAC (OFAC\_ID, OFAC\_ID\_Type, OFAC\_First\_Name, OFAC\_Middle\_Name, OFAC\_Last\_Name, OFAC\_DOB, OFAC\_Verbal\_Id, OFAC\_Gender, OFAC\_Segment\_Type, OFAC\_Occupation, OFAC\_Address, OFAC\_City, OFAC\_Date, OFAC\_Bank\_Code, OFAC\_Bank\_Account,



OFAC\_Province, OFAC\_Status, OFAC\_Zipcode, OFAC\_Swift\_Code, OFAC\_Remark, OFAC\_Married\_Status)  
 DHBI (DHBI\_ID, DHBI\_ID\_Type, DHBI\_Name, DHBI\_DOB, DHBI\_Gender, DHBI\_Address, DHBI\_City, DHBI\_Province, DHBI\_Zipcode, DHBI\_Bank\_Code, DHBI\_Segment\_Type, DHBI\_Date, DHBI\_Facility, DHBI\_Collectibility, DHBI\_Status, DHBI\_Verbal Id, DHBI\_Currency, DHBI\_Occupation, DHBI\_Remark, DHBI\_Outstanding, DHBI\_Status\_Nikah)  
 DHN (DHN\_ID, DHN\_ID\_Type, DHN\_Name, DHN\_DOB, DHN\_Gender, DHN\_Address, DHN\_City, DHN\_Province, DHN\_Zipcode, DHN\_Bank\_Code, DHN\_Segment\_Type, DHN\_Date, DHN\_Status, DHN\_Verbal Id, DHN\_Currency, DHN\_Occupation, DHN\_Remark, DHN\_Bank\_Account, DHN\_Cheque\_Number, DHN\_Cheque\_Amount)

BlackList\_Criminal, BlackList\_Type\_Blood, BlackList\_Married\_Status)

Diagram ER database daftar hitam sentralisasi PT.International Remittance Bank adalah :



Gambar 9. Diagram ER

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya.

Dalam kasus ini ada beberapa aturan :

1. Pada Tabel Daftar Hitam sentral (BlackList) yang merupakan gabungan dari tabel DHN, DHBI, AML, dan OFAC, diperlukan field sebagai identitas daftar hitam dari masing-masing sumber.
2. Agar menghindari terjadinya redudansi dan terpenuhi data integrity, maka pada tabel BlackList selain memasukkan field Id dan Id\_Type, maka diperlukan tambahan key yaitu Country\_Id dan BlackList\_Type dan diperlukan tambahan tabel master data untuk menyimpan country, jenis blacklist, dan tipe identitas tambahan yaitu BlackListCountry, BlackListType dan BlackListIDType .
3. Seseorang dalam satu negara bisa memiliki daftar hitam DHN, DHBI, AML, atau pun OFAC, dengan tipe identitas bermacam-macam dengan no identitas yang unik.

Jadi bentuk 2NF nya adalah :

BlackListCountry (BlackList\_Country\_ID, BlackList\_Country\_Description)  
 BlackListType (BlackList\_Type, BlackList\_Type\_description)  
 BlackListIDType (BlackList\_ID\_Type, BlackList\_ID\_Description)  
 BlackList (BlackList\_Country\_ID, BlackList\_Type, BlackList\_ID\_Type, BlackList\_Name, BlackList\_DOB, BlackList\_Gender, BlackList\_Address, BlackList\_City, BlackList\_Province, BlackList\_Zipcode, BlackList\_Bank\_Code, BlackList\_Segment\_Type, BlackList\_Date, BlackList\_Status, BlackList\_Verbal Id, BlackList\_Currency, BlackList\_Occupation, BlackList\_Remark, BlackList\_Bank\_Account, BlackList\_Cheque\_Number, BlackList\_Cheque\_Amount, BlackList\_Facility, BlackList\_Collectibility, Blacklist\_Status, BlackList\_Outstanding, BlackList\_Occupation, BlackList\_Religion, BlackList\_Swift\_Code,

4.2.5. Perancangan database secara fisik

Struktur data Tabel Daftar Hitam Sentral (BlackListCountry) :

Column Name	Data Type	Allow Nulls
BlackList_Country_ID	char(10)	<input type="checkbox"/>
BlackList_Country_De...	varchar(50)	<input checked="" type="checkbox"/>

Gambar 10. Struktur data Tabel Daftar Hitam Sentral

Struktur data tabel BalckListType

Column Name	Data Type	Allow Nulls
BlackList_Type	nchar(10)	<input type="checkbox"/>
BlackList_type_desc...	varbinary(50)	<input checked="" type="checkbox"/>

Gambar 11. Struktur data tabel BalckListType

Struktur data tabel BlackListIDType

Column Name	Data Type	Allow Nulls
BlackList_ID_Type	nchar(10)	<input type="checkbox"/>
BlackList_ID_Description	varchar(50)	<input checked="" type="checkbox"/>

Gambar 12. Struktur data tabel BlackListIDType

Struktur data Tabel Daftar Hitam Sentral (Tb.Blacklist) :

Column Name	Data Type	Allow Nulls
BLACKLIST_COUNTRY_ID	char(3)	<input type="checkbox"/>
BLACKLIST_TYPE	varchar(50)	<input type="checkbox"/>
BLACKLIST_ID_TYPE	char(2)	<input type="checkbox"/>
BLACKLIST_ID	char(10)	<input type="checkbox"/>
BLACKLIST_NAME	varchar(100)	<input checked="" type="checkbox"/>
BLACKLIST_DOB	datetime	<input checked="" type="checkbox"/>
BLACKLIST_GENDER	char(2)	<input checked="" type="checkbox"/>
BLACKLIST_VERBAL_ID	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_ADDRESS	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_ZIPCODE	char(10)	<input checked="" type="checkbox"/>
BLACKLIST_CITY	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_PROVINCE	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_TYPE_BLO...	char(2)	<input checked="" type="checkbox"/>
BLACKLIST_OCCUPAT...	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_MARRIED...	char(10)	<input checked="" type="checkbox"/>
BLACKLIST_SEGMENT...	char(2)	<input checked="" type="checkbox"/>
BLACKLIST_DATE	datetime	<input checked="" type="checkbox"/>
BLACKLIST_SWIFT_C...	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_BANK_CODE	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_ACCOUNT...	char(20)	<input checked="" type="checkbox"/>
BLACKLIST_CHEQUE...	char(10)	<input checked="" type="checkbox"/>
BLACKLIST_CHEQUE...	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_CURRENCY	char(3)	<input checked="" type="checkbox"/>
BLACKLIST_FACILITY	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_OUTSTAN...	varchar(MAX)	<input checked="" type="checkbox"/>
BLACKLIST_COLLECTI...	char(2)	<input checked="" type="checkbox"/>
BLACKLIST_RELIGION	char(10)	<input checked="" type="checkbox"/>
BLACKLIST_CRIMINAL	char(10)	<input checked="" type="checkbox"/>
BLACKLIST_REMARK	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_FILLER1	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_FILLER2	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_FILLER3	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_FILLER4	varchar(50)	<input checked="" type="checkbox"/>
BLACKLIST_FILLER5	varchar(50)	<input checked="" type="checkbox"/>

Gambar 13. Struktur data fisik

**4.2.6. Kapasitas Ukuran Penyimpanan**

Asumsi jumlah record pertama kali masuk ke tabel blacklist adalah 1juta record, dengan pertumbuhan pertahunnya adalah 10% dari total record pertamakali dan seterusnya, maka storage yang perlu disediakan adalah sebesar total record data ditambah dengan total record data.

Jumlah Byte dalam 1 record :  $1131 \times 8 = 9048$

1 juta record data + log = 2 (1juta x 9048)

Pertumbuhan data per tahun = 10 %

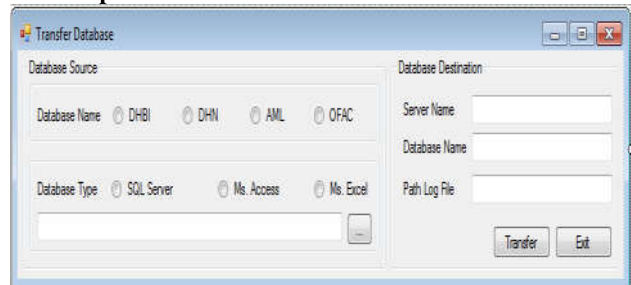
Jumlah Byte data dalam 5 tahun = 29.143.788.960 atau 27, 14gb.

Data akan dilakukan archive jika lebih dari satu tahun dengan menggunakan external storage.

**4.3. Pembuatan Prototipe**

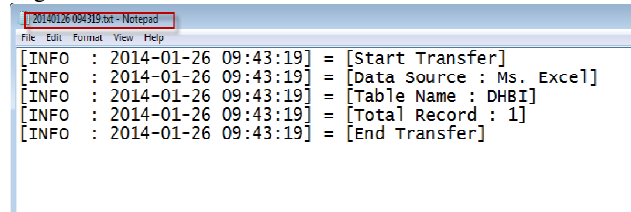
Pembuatan prototipe ini menggunakan Microsoft Visual Studio 2010 (.Net), prototipenya terdiri dari :

**4.3.1. Impor tabel daftar hitam**



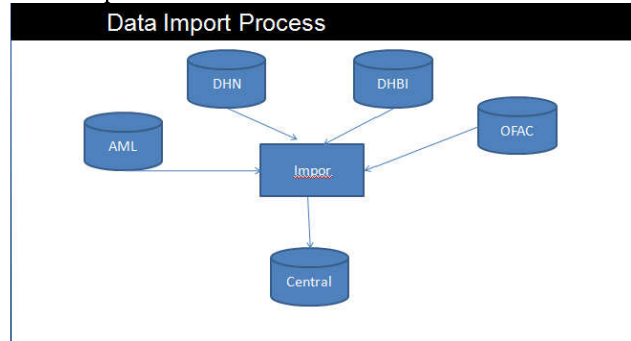
Gambar 14. Antarmuka Impor Database

**Log File**



Gambar 15. Antarmuka log file import database

**Proses impor database**



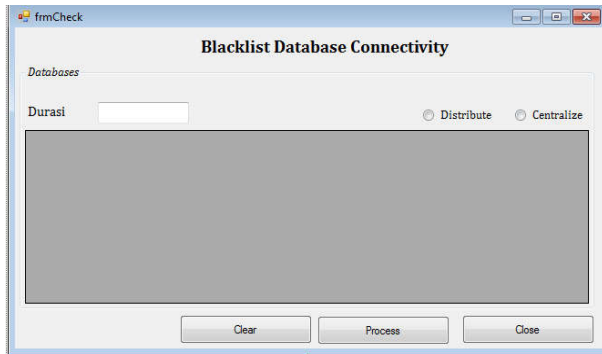
Gambar 16. Proses impor database

Algoritmanya adalah :

- Buka koneksi ke database DHBI, DHN, AML dan OFAC.
- Baca tabel DHN, ambil semua field nya, hard cord 'DHN' untuk tipe blacklist, 'IDR' untuk currency, dan 'ID' untuk country\_id.
- Baca tabel DHBI, ambil semua field, hard code 'DHBI' untuk tipe blacklist dan 'ID' untuk country\_id.
- Baca tabel AML, ambil semua field, hard code 'AML' untuk tipe blacklist.
- Baca tabel OFAC, ambil semua field, hard code 'OFAC' untuk tipe blacklist.
- Jika data sudah ada di tabel blacklist maka tulislah dalam log error, jika tidak ada
- Masukkan kedalam tabel blacklist dan tulis dalam log sukses.
- Export log file dalam bentuk file teks dengan naming file date time.



**4.3.2. Perbandingan kecepatan akses database daftar hitam terpisah dengan terpusat**



Gambar 17. Antarmuka perbandingan kecepatan akses

Algoritmanya :

Pilih option button databasenya

Jika option button = terpusat, maka koneksi ke database central

jika tidak maka koneksi ke beberapa database

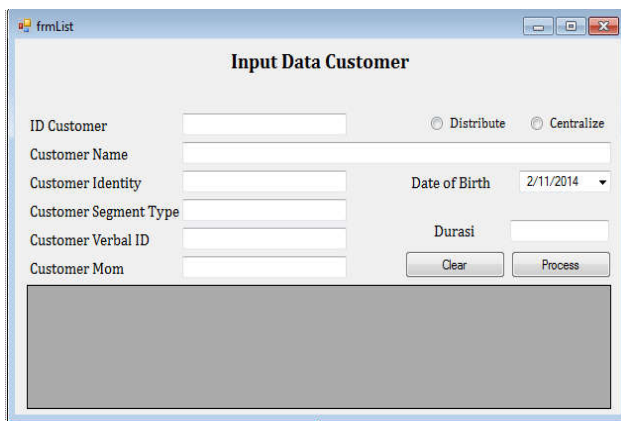
durasi = time awal

ambil record data nya

tampilkan di grid view

durasi = time akhir - time awal

**4.3.3. Validasi Daftar Hitam**



Gambar 18. Antarmuka validasi daftar hitam

Algoritmanya :

Pilih option button databasenya

Jika option button = terpusat, maka koneksi ke database central

jika tidak maka koneksi ke beberapa database

Input semua field

Tekan tombol process, system akan melakukan pencarian ke database

durasi = time awal

jika ada tampilkan di grid view

durasi = time akhir - time awal

**4.3.4. Pengujian Kualitas ISO 9126**

Kualitas perangkat lunak dapat dinilai melalui ukuran-ukuran dan metode-metode tertentu, serta melalui pengujian-pengujian software. Salah satu tolak ukur kualitas perangkat lunak adalah ISO 9126, yang dibuat oleh *International Organization for Standardization (ISO)* dan *International Electrotechnical Commission (IEC)*. ISO 9126 mendefinisikan kualitas produk perangkat lunak, model, karakteristik mutu, dan metrik terkait yang digunakan untuk mengevaluasi dan menetapkan kualitas sebuah produk software. Standar ISO 9126 telah dikembangkan dalam usaha untuk mengidentifikasi atribut-atribut kunci kualitas untuk perangkat lunak komputer. Faktor kualitas menurut ISO 9126 meliputi enam karakteristik kualitas sebagai berikut:

1. *Functionality*(Fungsionalitas). Kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Kehandalan). Kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Usability*(Kebergunaan). Kemampuan perangkat lunak untuk dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.
4. *Efficiency* (Efisiensi). Kemampuan perangkat lunak untuk memberikan kinerja yang sesuai dan relatif terhadap jumlah sumber daya yang digunakan pada saat keadaan tersebut.
5. *Maintainability*(Pemeliharaan). Kemampuan perangkat lunak untuk dimodifikasi.Modifikasi meliputi koreksi, perbaikan atau adaptasi terhadap perubahan lingkungan, persyaratan, dan spesifikasi fungsional.
6. *Portability*(Portabilitas). Kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.

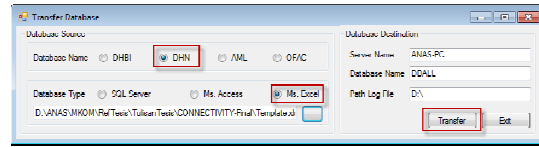
Pengujian ini dilakukan di Gedung PT.International Remittance Bank lantai 2 ruang UAT pada tanggal 20 Januari 2014, dengan para penguji :

Tabel 6. Skenario dan skrip Pengujian terdiri dari :

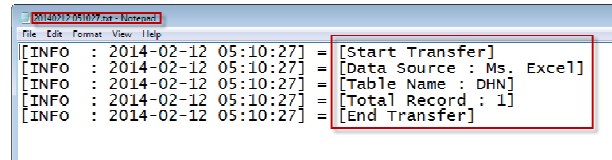
<b>Nama</b>	<b>Bagian</b>	<b>Lama bekerja</b>	<b>Pendidikan</b>
Ahmad Raihan	Database Administrator	1 tahun	S1
Hendri Wibawa	IT Payment Head	2 tahun	S1
Edwin Arifin	IT Business Process Improvement Head	3,4 tahun	S2

Tabel 7. Skenario dan skrip Pengujian terdiri

Date	Scenario	Test Script	Tester	Expected Result	Result		Notes	Sign Off
					OK	Not OK		
20.01.2014	Import Database	Impor dari tabel DHN	Ahmad Raihan	Impor berhasil dan file log berhasil dibuat				
		Cek Log File						
20.01.2014	Import Database	Impor dari tabel DHBI	Ahmad Raihan	Impor berhasil dan file log berhasil dibuat				
		Cek Log File						
20.01.2014	Import Database	Impor dari tabel DHBI	Ahmad Raihan	Impor berhasil dan file log berhasil dibuat				
		Cek Log File						
20.01.2014	Import Database	Impor dari tabel AML	Ahmad Raihan	Impor berhasil dan file log berhasil dibuat				
		Cek Log File						
20.01.2014	Import Database	Impor dari tabel DHN	Ahmad Raihan	Impor berhasil dan file log berhasil dibuat				
		Cek Log File						
20.01.2014	Benchmark Speed	Tampilkan Data di database terpisah	Hendri Wibawa	Data berhasil ditampilkan dan durasi waktu				
		Tampilkan Data di database terpusat	Hendri Wibawa	Data berhasil ditampilkan dan durasi waktu				
20.01.2014	Validasi Daftar Hitam	Imput data customer menggunakan database terpisah dan masukkan salah satu data blacklist	Edwin Arifin	Validasi berhasil dan durasi waktu ditampilkan				
20.01.2014		Imput data customer menggunakan database terpisah dan masukkan salah satu blacklist	Edwin Arifin	Validasi berhasil dan durasi waktu ditampilkan				



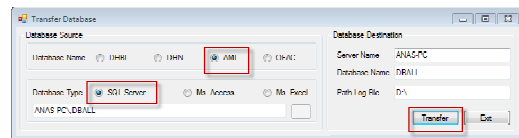
Gambar 22. Impor database DHN



Gambar 23. Log file

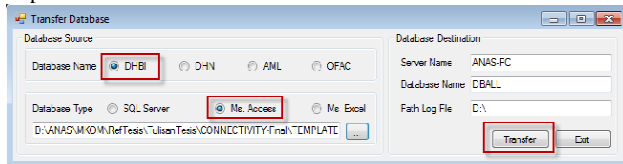
BLACKLIST_CO...	BLACKLIST_TYP...	BLACKLIST_ID...	BLACKLIST_ID...	BLACKLIST_NAM...	BLACKLIST_DO...	BLACKLIST_SE...	BLACKLIST_VE...	BLACKLIST_AD...	BLACKLIST_Z...
DHN	2	1	4	2011-01-01 00:...	6	5	7	8	
DHN	2	1	4	2012-02-02 00:...	6	5	7	8	

Gambar 24. Cek di tabel blacklist

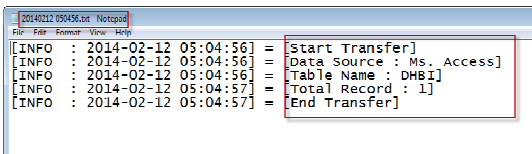


Gambar 25. Impor database AML

Hasil pengujian :  
Impor database DHBI dari Ms.Access



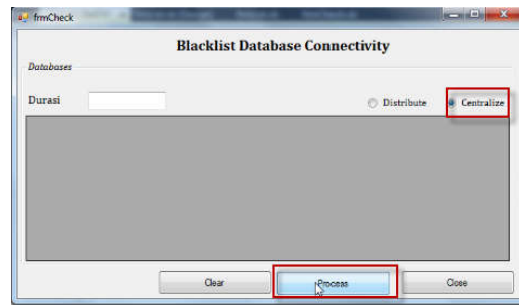
Gambar 19. Impor database DHBI dari Ms.Access



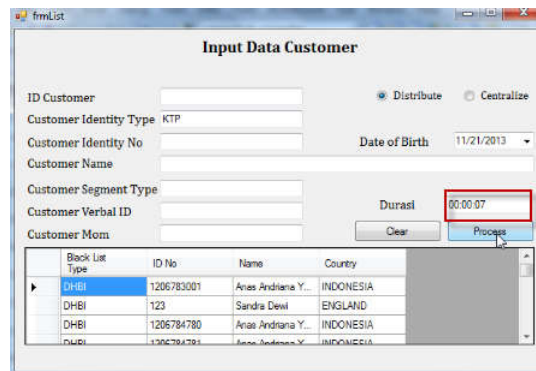
Gambar 20. Log file impor database

BLACKLIST_COUNTRY_ID	BLACKLIST_TYPE	BLACKLIST_ID_TYPE	BLACKLIST_ID	BLACKLIST_NAME	BLACKLIST_DO...	BLACKLIST_GENER...	BLACKLIST_VERBAL...
DHN	2	1	4	2011-01-01 00:00:00	6	5	

Gambar 21. Cek di tabel blacklist



Gambar 26. Perbandingan akses kecepatan database



Gambar 27. Form Input Data Customer

Date	Scenario	Test Script	Tester	Expected Result	Result	Notes	Sign Off
20.01.2014	Import Database	Import dari tabel Data	Ahmad Fauhan	Import berhasil dan file log berhasil dibuat	OK	Ben OK	
		Cek Log File			✓		
20.01.2014	Import Database	Import dari tabel Data	Ahmad Fauhan	Import berhasil dan file log berhasil dibuat	✓		
		Cek Log File			✓		
20.01.2014	Import Database	Import dari tabel Data	Ahmad Fauhan	Import berhasil dan file log berhasil dibuat	✓		
		Cek Log File			✓		
20.01.2014	Import Database	Import dari tabel Data	Ahmad Fauhan	Import berhasil dan file log berhasil dibuat	✓		
		Cek Log File			✓		
20.01.2014	Benchmark Speed	Tampilkan Data di database terpusat	Henri Wilansa	Data berhasil ditampilkan dan durasi waktu	✓		
		Tampilkan Data di database terpusat	Henri Wilansa	Data berhasil ditampilkan dan durasi waktu	✓		

Gambar 28. UAT Sign Off

20.01.2014	Validasi Daftar Hitam	Input data customer menggunakan database terpusat dan masukkan salah satu data blacklist	Edwin Arifin	Validasi berhasil dan durasi waktu ditunjukkan	✓		
		Input data customer menggunakan database terpusat dan masukkan salah satu blacklist	Edwin Arifin	Validasi berhasil dan durasi waktu ditunjukkan	✓		

Gambar 29. UAT Sign Off

Hasil Pengujian ISO 9126

Setelah melakukan UAT, selanjutnya para tester mengisi kuisisioner untuk menilai dari sisi functionality, reliability, usability dan efficiency terhadap sentralisasi database daftar hitam, berikut ini hasilnya :

Kuisisioner Responden 1

Black Customer Centralization Questionnaire  
 Nama Responden : Henri Wilansa  
 Jabatan : IT Payment Analyst  
 Tanggal : 22/01/2014

Isilah dengan tanda silang pada questionnaire dibawah ini :

Karakteristik	Sub-karakteristik	1	2	3	4	5
Functionality	Suitability					✓
	Accuracy					✓
	Security					✓
	Interoperability					✓
	Compliance					✓
Reliability	Maturity					✓
	Fault tolerance					✓
	Recoverability					✓
Usability	Understandability					✓
	Learnability					✓
	Operability					✓
	Attractiveness					✓
Efficiency	Time behavior					✓
	Resource behavior					✓
	Instalability					✓
	Coexistence					✓
	Replaceability					✓

Gambar 30. Kuisisioner Responden 1

Kuisisioner responden 2

Black Customer Centralization Questionnaire

Nama Responden : Edwin Arifin  
 Jabatan : IT - BPI Head  
 Tanggal : 20/01/2014

Isilah dengan tanda silang pada questionnaire dibawah ini :

Karakteristik	Sub-karakteristik	1	2	3	4	5
Functionality	Suitability					✓
	Accuracy					✓
	Security					✓
	Interoperability					✓
	Compliance					✓
Reliability	Maturity					✓
	Fault tolerance					✓
	Recoverability					✓
Usability	Understandability					✓
	Learnability					✓
	Operability					✓
	Attractiveness					✓
Efficiency	Time behavior					✓
	Resource behavior					✓
	Instalability					✓
	Coexistence					✓
	Replaceability					✓

Gambar 31. Kuisisioner Responden 2

Kuisisioner responden 3

Black Customer Centralization Questionnaire

Nama Responden : Ahmad Fauhan  
 Jabatan : Database Administrator  
 Tanggal : 20/01/2014

Isilah dengan tanda silang pada questionnaire dibawah ini :

Karakteristik	Sub-karakteristik	1	2	3	4	5
Functionality	Suitability					✓
	Accuracy					✓
	Security					✓
	Interoperability					✓
	Compliance					✓
Reliability	Maturity					✓
	Fault tolerance					✓
	Recoverability					✓
Usability	Understandability					✓
	Learnability					✓
	Operability					✓
	Attractiveness					✓
Efficiency	Time behavior					✓
	Resource behavior					✓
	Instalability					✓
	Coexistence					✓
	Replaceability					✓

Gambar 32. Kuisisioner Responden 3

Dari 3 responden tersebut diperoleh rata-rata penilaian terhadap sentralisasi database daftar hitam :

CHARACTERISTICS	SUB CHARACTERISTICS	RESPONDEN 1	RESPONDEN 2	RESPONDEN 3	AVERAGE
FUNCTIONALITY	SUITABILITY	5	4	5	4.666666667
	ACCURACY	5	5	5	5
	SECURITY	5	4	4	4.333333333
	INTEROPERABILITY	5	5	5	5
	COMPLIANCE	5	4	5	4.666666667
RELIABILITY	MATURITY	4	3	3	3.333333333
	FAULT TOLERANCE	3	4	4	3.666666667
	RECOVERABILITY	5	4	3	4
USABILITY	UNDERSTANDIBILITY	5	5	4	4.666666667
	LEARNABILITY	5	5	5	5
	OPERABILITY	4	5	4	4.333333333
EFFICIENCY	ATTRACTIVENESS	5	5	4	4.666666667
	TIME BEHAVIOR	4	4	5	4.333333333
	RESOURCE BEHAVIOR	5	5	5	5
	INSTABILITY	5	5	5	5
	COEXISTENCE	5	5	5	5
	REPLACEABILITY	5	5	5	5
	Total Score	80	77	76	77.666666667
	Ideal Score	85			
	Score Actual	91.37%		Sangat Baik	

Gambar 33. Hasil kuisioner

Hasil dari pengujian terhadap sentralisasi daftar hitam menggunakan ISO 9126 diperoleh prosentase 91.37%, sangat baik untuk diimplementasikan.

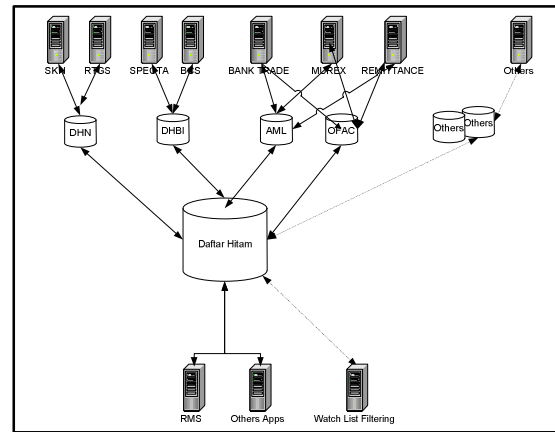
#### 4.4. Implikasi Penelitian

Berdasarkan pada penelitian yang telah dilaksanakan, maka beberapa implikasi yang akan terjadi apabila sentralisasi database daftar hitam diimplementasikan pada PT.International Remittance Bank, meliputi aspek sistem dan aspek manajerial.

##### 4.4.1. Aspek Sistem

Implementasi ini akan menjadi akan memberikan manfaat diantaranya :

1. Arsitektur dan desain menjadi efektif .
2. Mudah dalam mengelola dan mengawasi database daftar hitam.
3. Mudah dalam pengembangan aplikasi karena sudah ada standar format baik untuk pembuatan laporan maupun aplikasi pencegahan dini atau validasi yang terintegrasi terhadap transaksi yang mencurigakan (watch list filtering) dari semua daftar hitam yang ada dalam transaksi financial yang bersifat online transactional.
4. Menurunkan biaya investasi infrastruktur, jaringan dan sekuriti.



Gambar 34. Arsitektur Database

##### 4.4.2. Aspek Manajerial

Implementasi ini memberikan dampak yang baik terhadap PT.International Remittance Bank dalam aspek manajerial, diantaranya :

1. Mengurangi resiko operasional, kredit dan resiko reputasi.
2. Meningkatkan kepuasan kepada nasabah.
3. Mendukung pemerintah nasional maupun internasional dalam memberantas adanya transaksi yang mencurigakan baik *Anti Money Laundering (AML)*, pendanaan teroris maupun kejahatan kriminal lainnya, menjadikan PT.International Remittance Bank menjadi bank yang sangat peduli terhadap pengelolaan resiko sebagai implementasi dari *Good Corporate Governance*.

##### 4.5. Rencana Implementasi

Sentralisasi database daftar hitam ini rencananya akan diimplementasikan di PT.International Remittance Bank pada tahun 2014 pada tanggal 27 Maret 2014, dengan jadwal project sebagai berikut :

Task Name	Duration	Start
▣ Sentralisasi Database Daftar Hitam	60 days?	Mon 06/01/14
▣ Inisiasi	1 day	Mon 06/01/14
Kick Off Project	1 day	Mon 06/01/14
▣ Perencanaan	11 days	Tue 07/01/14
Perencanaan Pengadaan Hardware & Software	5 days	Tue 07/01/14
Pemilihan vendor Hardware & Software	5 days	Tue 14/01/14
PENETAPAN PENETAPAN PEMENANG LELANG	1 day	Tue 21/01/14
Pembuatan Requirement	5 days	Tue 07/01/14
▣ Eksekusi	51 days?	Tue 14/01/14
Analisa Requirement	1 day?	Tue 14/01/14
Pembuatan FSD	5 days	Wed 15/01/14
Pembelian Hardware & Software	2 days	Wed 22/01/14
Development & Unit Test	20 days	Wed 22/01/14
User Acceptance Test	20 days	Wed 19/02/14
Sosialisasi & Training	5 days	Wed 19/03/14
▣ Implementasi & Controlling	8 days	Wed 19/03/14
Review CCF	1 day	Wed 19/03/14
Setup Hardware & Deploy Application	5 days	Thu 20/03/14
Go Live	1 day	Thu 27/03/14
Closing Project	1 day	Fri 28/03/14

Gambar 35. Rencana Implementasi

Dari gambar 35 dapat diketahui bahwa project ini dimulai pada tanggal 6 Januari 2014 dan akan berakhir pada

28 Maret 2014, total durasinya adalah 60 hari kerja, untuk pembuatan database dan fungsi validasi daftar hitam dilakukan secara *in-house*, dan metodologi yang digunakan dalam perancangan database daftar hitam pada PT.International Remittance Bank adalah menggunakan prototipe dimana pihak Internal IT dan para pengguna terlibat penuh dalam proyek tersebut agar menghasilkan rancangan database dan aplikasi yang sesuai dengan kebutuhan dari para pengguna PT.International Remittance berdasarkan kebijakan IT (CMMI).

## 5. PENUTUP

### 5.1. Kesimpulan

Berdasarkan uraian pada bab-bab sebelumnya, maka Penulis dapat menyimpulkan bahwa database Daftar Hitam pada Internasional Remittance Bank dirancang secara sentralisasi dengan menggunakan teknik normalisasi yang menggabungkan dari database DHBI, DHN, AML, dan OFAC menjadi database daftar hitam yang tersentral. Perancangan database daftar hitam ini sudah diuji menggunakan pengujian kualitas menggunakan ISO9126 yang memperoleh hasil prosentase sebesar 91.37%, artinya sentralisasi database daftar hitam ini sangat baik untuk diimplementasikan pada PT.International Remittance Bank.

Sentralisasi database daftar hitam pada PT.International Remittance Bank ini akan memberikan manfaat baik dari aspek sistem maupun manajerial, yaitu :

1. Arsitektur dan desain menjadi efektif .
2. Mudah dalam mengelola dan mengawasi database daftar hitam.
3. Mudah dalam pengembangan aplikasi karena sudah ada standar format baik untuk pembuatan laporan maupun aplikasi pencegahan dini atau validasi yang terintegrasi terhadap transaksi yang mencurigakan (*watch list filtering*) dari semua daftar hitam yang ada dalam transaksi financial yang bersifat online transactional.
4. Menurunkan biaya investasi infrastruktur, jaringan dan sekuriti.
5. Mengurangi resiko operasional, resiko kredit, dan resiko reputasi.
6. Meningkatkan kepuasan terhadap nasabah-nasabah PT.International Remittance Bank.
7. Mendukung pemerintah nasional maupun internasional dalam memberantas adanya transaksi yang mencurigakan baik *Anti Money Laundering* (AML), pendanaan teroris maupun kejahatan kriminal lainnya, menjadikan PT.International Remittance Bank menjadi bank yang sangat peduli terhadap

### 5.2. SARAN

Implementasi Sentralisasi Database Daftar Hitam pada PT.Internasional Remittance ini akan dilaksanakan pada tanggal 27 Maret 2014, dengan total durasi proyek selama 60 hari kerja dan dikerjakan secara *in-house* oleh internal IT PT.Internasional Remittance Bank dengan menggunakan metode prototipe agar sesuai dengan kebutuhan dari para pengguna pada PT.International Remittance Bank.

Penulis menyarankan agar implementasi sentralisasi database daftar hitam pada International Remittance Bank ini dapat dilaksanakan sesuai dengan jadwal proyek yang sudah ditentukan sebelumnya untuk mengatasi segala permasalahan yang ada dan memberikan banyak manfaat sesuai dari kesimpulan diatas.

### DAFTAR PUSTAKA

- [1] "Prototyping Perangkat Lunak", Wikipedia, tanggal 16 Agustus 2013, [http://id.m.wikipedia.org/wiki/Prototyping\\_perangkat\\_lunak](http://id.m.wikipedia.org/wiki/Prototyping_perangkat_lunak) (Diakses 15 Februari 2014).
- [2] "Software Quality Assurance", Wikipedia, tanggal 17 Desember 2013, [http://en.m.wikipedia.org/wiki/Software\\_quality\\_assurance](http://en.m.wikipedia.org/wiki/Software_quality_assurance) (Diakses 15 Februari 2014).
- [3] "ISO 9126", Wikipedia, tanggal 6 April 2013, [http://id.m.wikipedia.org/w/index.php?title=ISO\\_9126&action=history](http://id.m.wikipedia.org/w/index.php?title=ISO_9126&action=history) (Diakses 15 Februari 2014).
- [4] H.M.Jogijanto, "Analisa dan Desain Sistem Informasi", PT.ANDI OFFSET, Jogjakarta, 1989.
- [5] Ali, Muhammad, "Materi Kuliah Basis Data", <http://ranykusnaningsih.blogspot.com/2009/02/teknik-perancangan-basis-data.html><http://teknik-informatika.com/perancangan-basis-data/www.google.com>, Jakarta, 1995 (Diakses 25 September 2013).
- [6] Ali, Muhammad, "Materi Kuliah Basis Data", <http://ranykusnaningsih.blogspot.com/2009/02/teknik-perancangan-basis-data.html><http://teknik-informatika.com/perancangan-basis-data/www.google.com>, Jakarta, 1995 (Diakses 25 September 2013).
- [7] Arif, "Basis Data Terdistribusi", <http://arifprof.blogspot.com/2013/05/basis-data-terdistribusi.html>, Jakarta, 01/05/2013 (Diakses 25 September 2013).