

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Strategi**

Strategi merupakan alat untuk mencapai tujuan, yang mana isu kritis atau faktor keberhasilan dapat dibicarakan, serta keputusan strategik bertujuan untuk membuat dampak yang besar serta jangka panjang kepada perilaku dan keberhasilan. Strategi pada hakikatnya adalah perencanaan (planning) dan manajemen (management) untuk mencapai suatu tujuan. Tetapi, untuk mencapai tujuan tersebut, strategi tidak berfungsi sebagai peta jalan yang hanya menunjukkan arah saja, melainkan harus mampu menunjukkan bagaimana taktik operasionalnya

##### **2.1.1 Definisi Strategi**

Menurut [17] Strategi merupakan suatu proses penentuan rencana para pemimpin puncak yang berfokus pada tujuan jangka panjang organisasi, disertai suatu penyusunan, cara atau upaya bagaimana agar tujuan tersebut dapat dicapai Menurut [18] Strategi adalah tujuan jangka panjang dari suatu perusahaan, serta pendayagunaan dan alokasi semua sumber daya yang penting untuk mencapai tujuan tersebut. Menurut [19] Strategi adalah suatu proses pengevaluasian kekuatan dan kelemahan perusahaan dibandingkan dengan peluang dan ancaman yang ada dalam lingkungan yang dihadapi dan memutuskan strategi pasar produk yang menyesuaikan kemampuan perusahaan dengan peluang lingkungan. Menurut [20] strategi adalah pola atau rencana yang mengintegrasikan tujuan, kebijakan dan aksi utama dalam hubungan yang kohesif. Suatu strategi yang baik akan membantu organisasi dalam mengalokasikan sumber daya yang dimiliki dalam bentuk unique berbasis kompetensi internal serta kemampuan mengantisipasi lingkungan. Menurut [21] strategi adalah merupakan serangkaian keputusan dan tindakan manajerial yang menentukan kinerja perusahaan dalam jangka panjang. Manajemen strategi meliputi pengamatan lingkungan, perumusan strategi (perencanaan strategis atau perencanaan jangka panjang). Implementasi strategi dan evaluasi serta pengendalian. Didalam strategi yang baik terdapat koordinasi tim kerja, memiliki tema, mengidentifikasi faktor pendukung yang sesuai dengan prinsip-prinsip pelaksanaan gagasan secara rasional, efisien dalam pendanaan dan memiliki taktik untuk mencapai tujuan secara efektif.

### 2.1.2 Konsep Strategi

- Perencanaan untuk semakin memperjelas arah yang ditempuh organisasi secara rasional dalam mewujudkan tujuan-tujuan jangka panjang.
- Acuan yang berkenan dengan penilaian konsistensi ataupun inkonsistensi perilaku serta tindakan yang dilakukan oleh organisasi.
- Sudut yang diposisikan oleh organisasi saat memunculkan aktivitasnya.
- Suatu perspektif yang menyangkut visi yang terintegrasi antara organisasi dengan lingkungannya yang menjadi batas bagi aktivitasnya.
- Rincian langkah taktis organisasi yang berisi informasi untuk mengelabui para pesaing.

## 2.2 Migrasi Database

### 2.2.1 Pengertian Migrasi Database

Jika dilihat dari maknanya migrasi dapat diartikan sebagai perpindahan dari suatu tempat ketempat lainnya, manusia yang berpindah dari kota satu ke kota lain, kawanan burung yang berpindah dari daerah yang bersuhu dingin ke daerah yang lebih hangat merupakan salah satu ilustrasi tentang apa yang disebut dengan migrasi. Namun Migrasi tidak hanya dikenal pada konteks makhluk hidup saja, konsep migrasi juga diterapkan pada bidang IT, migrasi data atau biasa dikenal dengan migrasi *database*, Merupakan salah satu penggunaan konsep migrasi yang digunakan di dunia IT. Migrasi data ialah suatu proses mentrasfer atau memindahkan data dari suatu lingkungan ke lingkungan lainnya, dimana lingkungan dapat diartikan sebagai aplikasi *database*, *platform* teknologi, lokasi penyimpanan data yang baru atau berbeda. Sebagai ilustrasi seorang database administrator ingin memindahkan data yang ia miliki dari aplikasi basis data Oracle menuju SQL server karena suatu alasan, maka proses pemindahan yang dilakukan oleh database administrator tadi merupakan proses migrasi data

### 2.2.2 Alasan Migrasi Database

Beberapa alasan yang mendasar organisasi untuk melakukan proses migrasi *database* seperti:

- Sistem yang baru dikembangkan mengharuskan suatu organisasi menggunakan aplikasi *database* yang baru.
- Teknologi yang ada di *database* lama sudah terlalu lama atau tua.

- Jika dilihat dari aspek bisnis bisa jadi basis data yang lama memerlukan biaya operasional dan perawatan yang jauh lebih besar daripada suatu organisasi harus *mengupgrade* ke sistem atau basis data yang baru.
- Sistem yang lama sudah tidak lagi efisien untuk mendukung proses bisnis disuatu organisasi yang semakin dinamis.

### 2.2.3 Tantangan Migrasi *Database*

Tantangan yang dihadapi oleh seorang database administrator sebagai eksekutor ketika melakukan migrasi data adalah bagaimana agar data-data lama yang akan dimigrasi tidak berubah sedikit pun ketika telah dipindah ke sistem yang baru selain itu ada beberapa aspek lain yang mempengaruhi kesuksesan migrasi data dari sistem lama ke sistem yang baru diantaranya adalah:

- Aspek data, apabila kompleksitas data yang akan dipindah itu tinggi baik dari sisi arsitektur, maupun struktur datanya maka proses migrasi yang akan dilakukan pun semakin sulit dan berakibat pada tingkat keberhasilan migrasi suatu data
- Aspek Aplikasi, Keberhasilan migrasi data juga bergantung dari seberapa besar perbedaan *platform*, sistem, serta standar dari sistem yang baru, semakin banyak perbedaan antara sistem baru dan sistem lama maka akan mempersulit proses migrasi data yang dapat meningkatkan potensi kegagalan ketika migrasi.
- Aspek Teknologi, pemetaan data menjadi lebih sulit jika bentuk serta topologi dari perangkat keras dan jaringan semakin tersebar sesuai geografis atau fungsi bisnis.
- Aspek Manusia, semakin banyak unsur manusia yang terlibat pada aktifitas pemasukan, pengorganisasian, pemeliharaan, dan pengawasan data, maka akan meningkatkan potensi terjadinya kesalahan yang berdampak pada kualitas data yang disimpan, juga memperbesar risiko kesalahan yang terjadi ketika proses migrasi.
- Aspek kebijakan, jika suatu perusahaan tidak memiliki standar yang jelas sebagai acuan dalam proses pengolahan data, maka akan mempersulit proses penentuan strategi migrasi yang tepat.

## 2.2.4 Jenis Strategi Migrasi Database

Jenis migrasi Database yang paling umum adalah sebagai berikut:

### 2.2.4.1 Big bang Migration Strategi

Yaitu semua data dipindahkan dalam sekali jalan dan sistem warisan segera dinonaktifkan [4].

### 2.2.4.2 Phased Migration Strategi

Yaitu data dipindahkan dalam bagian yang terpisah, mungkin secara geografis atau berdasarkan fungsi bisnis [4].

### 2.2.4.3 Parallel Migration Strategi

Yaitu data dipindahkan, tetapi sistem warisan diizinkan untuk berjalan, selama beberapa periode untuk menjamin bahwa target beroperasi dengan benar

### 2.2.4.4 Always Up Migration Strategi

Yaitu migrasi yang sistem sumber tidak boleh dimatikan (seperti sistem telepon) bahkan untuk sesaat, ini memerlukan alat migrasi spesial dengan setidaknya sinkronisasi yang handal dan perpindahan yang kompleks dari sistem warisan ke target [4].

## 2.2.5 Teknik Migrasi Database

Untuk menerapkan teknologi *database* progresif, industri perangkat lunak menghadapi tantangan memigrasikan database dan aplikasi. demikian juga beberapa teknik migrasi *database* ada di industri perangkat lunak yang dapat menyajikan informasi tentang *database* konversi dan aplikasi. Tabel berikut menunjukkan survei literatur tentang strategi model migrasi *database*:

**Table 2.1 Model Teknik Migrasi**

<b>Author</b>	<b>Database Migration Model Strategies</b>
Fishman et al., (1987)	Object Oriented Database Management System
Hardwick and Spooner. (1989)	Using Object Technology to Engineering Applications
Wilkinson et al., (1990)	Object Oriented Data Management System
Hainaut, J. (1991)	Database Reverse Engineering Models and Techniques
Crowe. M. K. (1993)	Object System over Relational Databases
Andreas Meier et al., (1994)	Hierarchical to Relational Database
Ian Graham. (1995)	Relational to Object Technology
Monk, S. et al., (1996)	Relational to Object Oriented Database
Andreas Behm et al., (1997)	Relational to Object Oriented System
Yury B.and Jens H. Jahnke (2001)	Legacy Databases to XML
Fong, J. and Cheung (2005)	Relational to XML Database
Abdel Salam Maatuk. Et al., (2008)	Relational to Object Oriented Relational Database (ORDBs)

Pendekatan-pendekatan ini dikategorikan ke dalam tiga dasar strategi yang terkait dengan migrasi *database*. Yang pertama adalah untuk penanganan data yang disimpan dalam *database* melalui antarmuka *XML*. Yang kedua adalah menghubungkan sumber relasional yang ada di *database* sumber ke sistem database target konseptual yang berbeda. Ketiga adalah memigrasikan sepenuhnya skema dan data di sumber *database* ke sistem *database* target yang setara. Itu strategi pertama dan kedua hanya berurusan dengan skema terjemahan. Keuntungan dari pendekatan pertama adalah bahwa data relasional masih dapat diakses sebagai *database* relasional.

Kerugiannya adalah ketidak efisienan karena telah menerjemahkan *Data Manipulation Language (DML)* perintah antara keduanya lapisan. Strategi kedua adalah untuk menerapkan lebih dari satu migrasi, bukan sekadar *overlay* sebuah antarmuka. Di dalam kasus, teknologi relasional bermigrasi ke *object*. Langkah yang paling penting dalam proses ini adalah untuk mendapatkan *object* yang berorientasi skema dari skema relasional dari sumber yang ada di sistem. Karena banyak investasi yang cukup besar dalam database relasional tradisional, bagian dari data mereka mungkin perlu diformat dan diimplementasikan dalam platform baru yang berbeda. Oleh karena itu, membangun antarmuka gateway antara kedua database mungkin lebih disukai. Migrasi ke DBMS baru mungkin merupakan keputusan yang baik jika sistem yang ada juga mahal untuk dipelihara.

Menurut [11] Ada banyak metrik yang dapat mengukur efektivitas dan efisiensi migrasi database yaitu sebagai berikut:

- Jumlah penyesuaian online diperlukan
- Persentase catatan yang dimigrasikan
- Persentase tabel yang dimigrasikan
- Persentase catatan yang dimigrasikan oleh teknologi
- Persentase catatan yang dimigrasikan menurut aplikasi
- Persentase data dengan masalah kualitas
- Jumlah kesalahan migrasi
- Dampak migrasi pada ukuran basis data
- Waktu Henti karena migrasi
- Diperlukan penyimpanan pementasan / perangkat keras
- Persentase kesalahan rekonsiliasi

## 2.3 TOGAF

TOGAF merupakan kepanjangan dari *The Open Group Architecture Framework* yang dikembangkan oleh *The Open Group Architecture Framework* pada tahun 1995. adalah *framework* arsitektur yang memberikan metode dan *tools* untuk membantu dalam penerimaan, produksi, penggunaan, dan perawatan arsitektur *enterprise*. TOGAF didasarkan pada model proses iteratif yang didukung oleh *best practice* dan sekumpulan aset arsitektur yang bisa digunakan kembali [5]. TOGAF sudah dikembangkan oleh lebih dari 300 perusahaan anggota forum arsitektur yang berasal dari perusahaan dan organisasi besar. Mengembangkan dan mendukung arsitektur *enterprise* merupakan teknik yang rumit karena mencakup berbagai *stakeholder* dan proses pengambilan keputusan di dalam organisasi. TOGAF memiliki peran penting di dalam standarisasi dan menjelaskan proses pengembangan arsitektur. TOGAF memberikan praktik kerangka kerja terbaik untuk menambah nilai, dan memungkinkan organisasi untuk membangun solusi yang dapat dilakukan lebih ekonomis dan mengakomodasi persoalan dan kebutuhan bisnis. Pada awalnya TOGAF digunakan oleh Departemen Pertahanan Amerika Serikat, namun pada perkembangannya banyak digunakan pada berbagai bidang seperti industri manufaktur, perbankan, pendidikan dan lain sebagainya. TOGAF digunakan untuk mengembangkan arsitektur *enterprise*, dimana terdapat metode dan alat yang detail untuk mengimplementasikannya. Hal inilah yang membedakan dengan kerangka kerja arsitektur *enterprise* yang lain. Salah satu kelebihan dari kerangka kerja ini adalah sifatnya fleksibel dan *open source* [6]. TOGAF adalah salah satu metode yang paling banyak diterima untuk mengembangkan arsitektur perusahaan. TOGAF memberikan metode yang rinci dalam membangun dan mengelola serta mengimplementasikan arsitektur *enterprise* dan sistem informasi yang disebut dengan *Architecture Development Method (ADM)* [16]. ADM merupakan metode generik yang berisikan sekumpulan aktifitas yang digunakan dalam memodelkan pengembangan arsitektur *enterprise*. Metode ini juga bisa digunakan sebagai panduan atau alat untuk merencanakan, merancang, mengembangkan dan mengimplementasikan arsitektur sistem informasi untuk organisasi [5]. TOGAF ADM merupakan metode yang fleksibel yang dapat mengidentifikasi berbagai macam teknik pemodelan yang digunakan dalam perencanaan, karena metode ini bisa disesuaikan dengan perubahan dan kebutuhan selama perancangan dilakukan. TOGAF ADM juga menyatakan visi dan prinsip yang

jelas tentang bagaimana melakukan pengembangan arsitektur *enterprise*, prinsip tersebut digunakan sebagai ukuran dalam menilai keberhasilan dari pengembangan arsitektur *enterprise* oleh organisasi.

### 2.3.1 Prinsip *Enterprise*

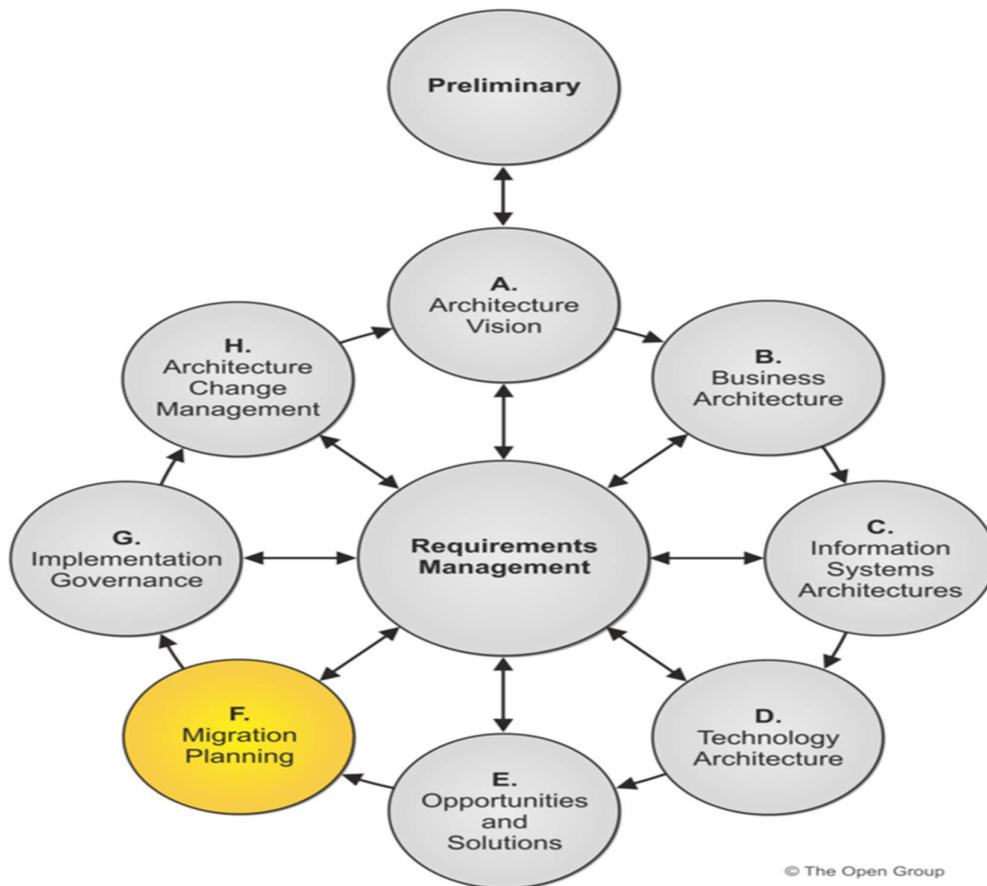
Pengembangan arsitektur yang dilakukan diharapkan mendukung seluruh bagian organisasi, termasuk unit-unit organisasi yang membutuhkan.

### 2.3.2 Prinsip Teknologi Informasi

Lebih mengarahkan konsistensi penggunaan teknologi informasi pada seluruh bagian organisasi, termasuk unit- unit organisasi yang akan menggunakan.

### 2.3.3 Prinsip Arsitektur

Merancang arsitektur sistem berdasarkan kebutuhan proses bisnis dan bagaimana mengimplementasikannya.



Gambar 2.1 Proses Pengembangan TOGAF ADM [5]

TOGAF terdiri atas 8 (delapan) fase yang berbentuk siklus (*cycle*). Fase-fase dalam metode TOGAF dapat dilihat pada gambar 2.1. Tahapan dari TOGAF ADM dapat dijelaskan sebagai berikut:

1) *Preliminary:*

Tahapan persiapan (*Preliminary Stage*) merupakan tahapan untuk menentukan ruang lingkup arsitektur *enterprise* yang akan dikembangkan serta menentukan komitmen dengan manajemen dalam pengembangan arsitektur *enterprise*.

2) *Architecture Vision:*

Menciptakan keragaman pandangan mengenai pentingnya arsitektur *enterprise* untuk mencapai tujuan organisasi yang dirumuskan dalam bentuk strategi serta menentukan ruang lingkup dari arsitektur yang akan dikembangkan. Pada tahapan ini berisikan kebutuhan-kebutuhan berkenaan dengan perancangan arsitektur sistem informasi yaitu profil organisasi, pendefinisian visi dan misi, tujuan organisasi, sasaran organisasi, proses bisnis organisasi, unit organisasi dan kondisi arsitektur saat ini.

3) *Business Architecture:*

Mendefinisikan kondisi awal arsitektur bisnis, menentukan model bisnis atau aktivitas bisnis yang diinginkan berdasarkan skenario bisnis. Pada tahapan ini *tools* dan *method* umum untuk permodelan seperti: *Unified Modeling Language* (UML) bisa digunakan untuk membangun model yang diperlukan.

4) *Information System Architecture :*

Pada tahapan ini lebih menekankan pada aktivitas bagaimana arsitektur sistem informasi dikembangkan. Pendefinisian arsitektur sistem informasi dalam tahapan ini meliputi arsitektur data dan arsitektur aplikasi yang akan digunakan dalam organisasi. Arsitektur data lebih memfokuskan bagaimana data digunakan untuk kebutuhan fungsi bisnis, proses dan layanan. Teknik yang bisa digunakan yaitu *ER-Diagram*, *Class Diagram*, dan *Object Diagram*.

5) *Opportunities and Solution:*

Mengevaluasi dan memilih alternatif solusi, identifikasi parameter strategis penilaian keterkaitan biaya dan manfaat, mendefinisikan strategi implementasi dan rencana implementasi. Pada tahapan ini lebih menekankan

pada manfaat yang diperoleh dari arsitektur *enterprise* yang meliputi arsitektur bisnis, arsitektur data, arsitektur aplikasi dan arsitektur teknologi, sehingga menjadi dasar bagi *stakeholder* untuk memilih dan menentukan arsitektur yang akan diimplementasikan.

6) *Migration Planning:*

Pada tahap ini lebih menekankan pada manfaat yang diperoleh dari arsitektur *enterprise* yang meliputi arsitektur bisnis, arsitektur data, arsitektur aplikasi dan arsitektur teknologi, sehingga menjadi dasar *stakeholder* untuk memilih dan menentukan arsitektur yang akan diimplementasikan.

7) *Implementation Governance:*

Pada tahap ini menyusun rekomendasi untuk pelaksanaan tatakelola implementasi yang sudah dilakukan, tatakelola yang dilakukan meliputi tata kelola organisasi, tata kelola teknologi informasi, dan tata kelola arsitektur.

8) *Management Change Architecture:*

Pada tahap ini menetapkan rencana manajemen arsitektur dari sistem yang baru dengan cara melakukan pengawasan terhadap perkembangan teknologi dan perubahan lingkungan organisasi, baik internal maupun eksternal serta menentukan apakah akan dilakukan siklus pengembangan arsitektur *enterprise*.

## **2.4 Peluang dan Solusi Perencanaan Migrasi**

Pada TOGAF Fase E dan F melihat penjadwalan dan pengorganisasian implementasi arsitektur baru. Penekanan ditempatkan pada membangun jalur migrasi, yang harus membawa manfaat bisnis sejati untuk setiap fase. Selama fase E, hasil fase elaborasi (B, C, dan D) dikonsolidasikan: arsitektur, persyaratan, dan kesenjangan. Konsolidasi ini merupakan bahan baku yang digunakan untuk menentukan arsitektur transisi, sambil mengingat kemampuan perusahaan untuk perubahan (misalnya, aplikasi baru untuk mengembangkan dan evolusi aplikasi yang ada, sesuai dengan cakupan fungsi bisnis). Kelayakan teknis dan organisasi, kompromi antara persyaratan dan biaya, dan kendala integrasi juga dipelajari. Fase F secara tepat menetapkan penjadwalan migrasi, serta konstitusi proyek implementasi dan organisasi, tujuan, dan biaya dari perencanaan migrasi.

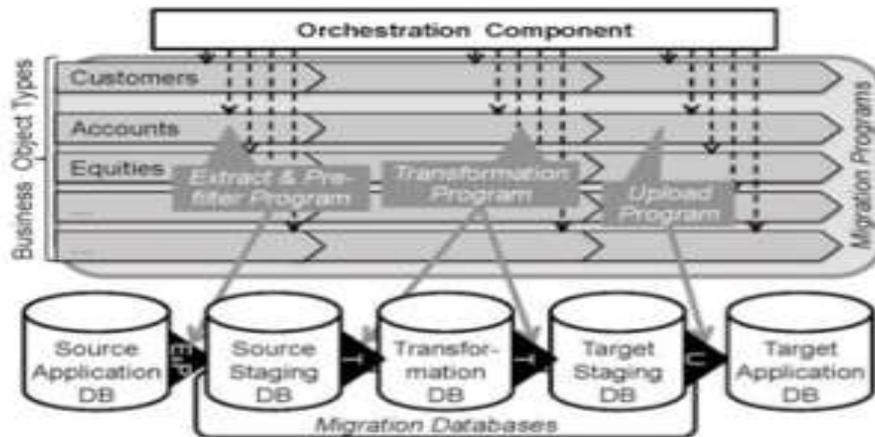
## 2.5 Arsitektur Migrasi Data

Semua proyek migrasi data menerapkan program migrasi, skrip dan aturan transformasi. Program menangani dan memodifikasi data, misalnya rekening tabungan atau pesanan mobil. Mereka berbeda dari sektor industri ke sektor industry lainnya, dari perusahaan ke perusahaan, dan dari proyek ke proyek. Namun demikian, mereka berlandaskan pada arsitektur migrasi data generik. Untuk alasan ini, ide dan prinsip yang mendasari melalui proyek - dan bahkan teknologi dan kerangka kerja konkret - dapat digunakan kembali dalam proyek di semua sektor industri [9]. Pertama, struktur umum untuk program migrasi adalah umum. Begitu juga konsep komponen dari (tiga) *database* migrasi data (Gambar 2.2). adalah sebagai berikut:

- *Database* sumber menjalankan penyimpanan salinan aplikasi *database* sumber untuk memisahkan kedua *database*. Ini mencegah proyek migrasi data dari kerusakan data atau memperlambat *database* pada aplikasi sumber yang masih digunakan.
- *Database* transformasi menyimpan hasil antara dari program migrasi data. Program migrasi *merge*, membandingkan, menggabungkan, memfilter, dan mengubah besar jumlah data. Ini seharusnya tidak pernah diterapkan di Bahasa pemrograman *Java* atau bahasa serupa. *Database* dioptimalkan untuk menangani kumpulan data besar secara efisien, misalnya kapan akan dieksekusi operasi di memori utama dan kapan harus menulis data di disk.
- *Database* target menyimpan hasil transformasi yang siap untuk aplikasi target unggah. Itu tergantung pada aplikasi target apakah ada unggahan pada API atau apakah data harus disalin langsung ke *tabel database*. Dalam kasus terakhir, unggah struktur *Database staging* berisi *tabel* yang setara dengan *database* aplikasi target. Jika ada unggah API, *tabel* di area *staging* target berisi parameter untuk menggunakan API. Tentu saja, *database* di atas dapat disimpulkan sebagai contoh yang logis. Dengan demikian, satu *instance database* fisik dapat menyimpan ketiganya yang menggunakan *logic*, misalnya, tiga skema berbeda. Istilah program migrasi data mengacu pada program memindahkan data dari *database* aplikasi sumber ke aplikasi database target dalam mentransformasikan data dan merepresentasi sesuai kebutuhan. Migrasi data dalam istilah program tunggal menggolongkan tiga program yang diperlukan untuk memigrasi satu

jenis *object* bisnis: ekstrak & pra-filter program, program transformasi, dan program unggahan.

- Sekalipun programnya spesifik untuk proyek, mereka tetap mengikuti *generic* struktur, Ini memudahkan pengaturan proyek dan berurusan dengan perubahan. Dua dimensi mencirikan program: tahapan dan jenis *object* bisnis. Tahap pertama adalah pra-filtering dan ekstraksi. Pra-filter & dan *ekstrak* program menyalin data yang relevan dari *database* aplikasi sumber. Data tetap tidak tersentuh oleh tingkat kemungkinan terbesar. Pra-filter berarti mengurangi data tetapi tidak memutuskan data mana yang harus disalin ke target *database* aplikasi. pengecualian akun ditutup beberapa dekade yang lalu. Namun, akun internal tertentu tidak akan difilter di fase pra-filter. Ini membutuhkan analisis mendalam untuk memutuskan apakah mereka diperlukan dalam aplikasi target. Memperkaya data dengan data yang awalnya tidak disimpan di sumber, tetapi diperlukan oleh aplikasi target [8], [9] dapat dilakukan sekarang atau dalam langkah selanjutnya, program transformasi. program transformasi berisi semua pemetaan data antara sumber dan struktur target. Misalnya, ID jenis akun untuk akun tabungan mungkin 4553 di aplikasi sumber dan 85571 di target. Program transformasi juga melakukan penyaringan *granular* untuk memutuskan data mana dari kebutuhan aplikasi target. Program ketiga dan terakhir adalah mengunggah program. Ini memuat data ke dalam aplikasi target *database*. Jika disediakan, program unggahan dapat menggunakan API atau menulis data langsung ke *table database* target. Menjalankan semua program migrasi data dalam urutan yang benar adalah sebuah tantangan. Jadi, Orkestrasi komponen memastikan urutan awal yang benar dari menggunakan mekanisme program seperti gambar dibawah ini.



Gambar 2.2 Arsitektur Migrasi Data [7]

## 2.6 Model Risiko Migrasi Data

Proyek migrasi data seringkali sangat terlihat di dalam organisasi. Para penulis sering memperhatikan ketakutan laten bahwa beralih ke aplikasi target atau migrasi data mungkin gagal. Rumah risiko migrasi data kami (Gambar 2.3) membantu untuk beralih dari ketakutan yang tak terduga akan risiko nyata. Ini melengkapi proyek *generic* metodologi manajemen migrasi data dengan risiko yang spesifik. Rumah risiko migrasi data memiliki tiga tingkatan: tingkat bisnis, tingkat manajemen TI, dan tingkat migrasi data. Berikut ini, setiap level dijelaskan dan didukung oleh contoh-contoh. Tingkat atas mengandung risiko bisnis, seringkali diartikulasikan oleh pelanggan. Tiga risiko bisnis yang paling relevan adalah:

### 2.6.1 Profitabilitas

Adalah tentang biaya dan pendapatan. Ini mencakup, pertama, biaya langsung. Biaya proyek migrasi data itu sendiri harus melekat pada anggaran proyek. Kedua, dari itu dapat berupa biaya tidak langsung dari transformasi data yang salah. Misalnya, produsen mobil mungkin memesan 500 mobil berwarna merah dan 20.000 mobil berwarna hitam. Jika, setelah migrasi, ada pesanan untuk 20.000 mobil berwarna merah dan 500 mobil berwarna hitam, maka ribuan mobil dirakit secara salah. Ketiga, ada juga efek lanjutan. Proyek migrasi data sebagai bagian dari proyek transisi sistem dapat menjadi prasyarat untuk penawaran baru, biaya operasi lebih rendah, atau pendapatan yang bias lebih tinggi.

### **2.6.2 Reputasi**

Jika misalnya bank mengabaikan untuk melakukan migrasi data yang bahkan memperbaiki kesalahan ini tidak mahal. Tetapi jika *performance* ternyata menghasilkan media yang *negative*, ini dapat menghambat reputasi bank.

### **2.6.3 Regulasi**

Di sektor-sektor tertentu, misalnya, industri keuangan, regulator adalah institusi yang harus dihormati dan mungkin ditakuti. Mereka tidak boleh mendapatkan kesan bahwa Sistem TI tidak dikelola dengan baik. Jika tidak, hukuman *moneter* bukanlah bentuk hukuman yang paling berat. Kelompok risiko kedua terdiri dari risiko manajemen TI yang penting. Mereka memiliki fokus teknis dan berada pada posisi tingkat tinggi.

### **2.6.4 Kehilangan Data atau Informasi.**

Migrasi data harus dapat memindahkan semua data yang relevan dari aplikasi sumber ke aplikasi target tanpa mengubah struktur awalnya.

### **2.6.5 Menargetkan Stabilitas Aplikasi**

Bahkan jika semua data yang relevan dipindahkan ke dalam aplikasi target, masih ada risiko bahwa data yang dimigrasi membahayakan stabilitas aplikasi target, misalnya, jika tidak memenuhi asumsi implisit tentang struktur data.

### **2.6.6 Kegagalan *Cut-over***

Pemutusan tidak berhasil karena kesalahan migrasi data. Ini bisa dibatalkan dengan program sedang berjalan. Atau, program-program berjalan dengan sukses, tetapi data tidak memungkinkan untuk beralih ke aplikasi target.

### **2.6.7 Waktu Henti yang Panjang**

*Cut-over* membutuhkan lebih banyak waktu daripada rencana awal. Misalnya, jalur produksi atau sistem operasional tidak tersedia lebih lama dari yang diharapkan. Dengan demikian, operasional tidak dapat menyediakan layanannya dan jalur produksi tidak dapat dijalankan.

### **2.6.8 Pembengkakan Anggaran Proyek**

Departemen TI dan manajer TI memiliki anggaran dan harus melaporkan anggaran secara actual. Tentu saja, biaya proyek yang lebih tinggi membawanya ke situasi yang tidak nyaman.



**Gambar 2.3 Rumah Risiko Migrasi Data [7]**

### 2.6.9 Penundaan

Penundaan proyek sederhana (tanpa biaya tambahan), bisa menjadi risiko. Penundaan dapat mengakibatkan, misalnya, dalam tidak tersedianya staf inti untuk lisensi yang habis masa berlakunya aplikasi lama, atau ketidakcocokan antar muka jika pajak otoritas atau bursa efek menutup antarmuka lama. Karena pengalaman kami dari proyek terbukti, manajer proyek migrasi data membutuhkan kepercayaan dari bisnis dan TI manajer. Karena itu, ia harus siap menjawab pertanyaan dibesarkan oleh bisnis mengenai bisnis dan manajemen TI risiko. Namun, ini adalah risiko abstrak. Migrasi data manajer proyek tidak dapat mengatasinya secara langsung. Sebaliknya, dia harus menunjukkan bagaimana bisnis dan manajemen TI berisiko memetakan secara konkret risiko proyek migrasi data. Lalu, dia bisa menunjukkan bagaimana dia mengatasi proyek migrasi data berisiko dia alamat dan bagaimana mereka memitigasi risiko tingkat yang lebih tinggi. Risiko proyek migrasi data terbagi dalam empat kategori. Program risiko migrasi data mencakup semua risiko yang terkait dengan program migrasi data. Daftar berikut menggambarkan yang paling penting:

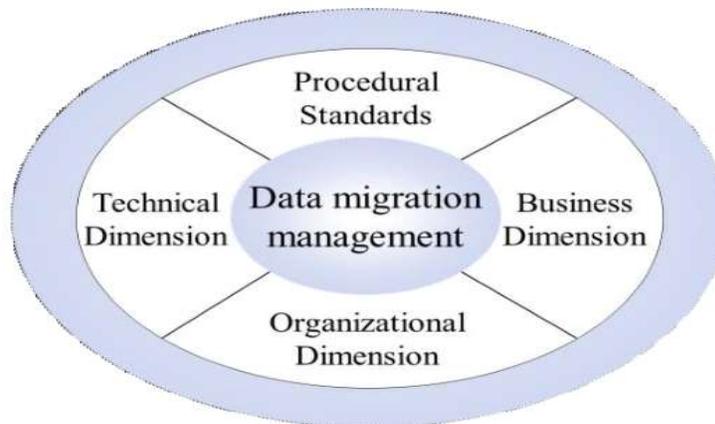
- Risiko kelengkapan mencakup risiko kehilangan satu atau lebih banyak *object* bisnis selama migrasi data. Ini juga mengambil risiko bisnis tambahan *object* (seperti pesanan untuk mobil biru) muncul di jendela aplikasi target yang tidak ada di sumber aplikasi.

- Risiko semantik dapat muncul jika *object* bisnis telah dimigrasikan (mis., pesanan mobil atau penyimpanan simpanan akun), tetapi semantik telah berubah. Pesanan untuk mobil biru, tapi sekarang mobil merah. Bisa juga bahwa nilainya tetap sama tetapi unit perubahan. Aplikasi sumber mungkin berfungsi dengan USD, targetnya satu jika EUR. Jadi 500 USD harus tidak menjadi 500 EUR.
- Risiko korupsi data berarti bahwa data yang dimigrasi tidak mencerminkan model data aplikasi target. Ini bisa terjadi jika ada kendala pada tingkat aplikasi (mis., yang hanya dipesan pada rekening tabungan dimungkinkan yang tidak menghasilkan tingkat akun negatif) yang tidak diberlakukan oleh basis data. Basis data tidak mencegah pemuatan dari data 'rusak', tetapi aplikasi mungkin macet nanti pada karena ini.
- Risiko stabilitas adalah risiko yang dihasilkan dari kesalahan pengkodean dalam kerusakan atau pembatalan migrasi data yang tidak diinginkan program.
- Risiko menjalankan migrasi merujuk pada pelaksanaan set lengkap program migrasi data. Ada dua jenis:
  - Risiko waktu eksekusi mencerminkan kemungkinan data itu program migrasi berjalan lebih lama dari yang diharapkan. Ini dapat terjadi misalnya jika semua tes dilakukan dengan subset data. Jika 10% dari data perlu satu jam untuk dimigrasi, itu tidak berarti bahwa 100% selesai dalam sepuluh jam. Jika menyortir atau bergabung, terlibat asumsi linearitas mungkin tidak berlaku lagi. Jika database dapat memproses 10% dari data di utama memori tanpa akses disk, tidak ada jaminan bahwa database juga dapat menangani 100% data dalam memori utama.
  - Risiko orkestrasi mengatasi tantangan dalam mengoordinasikan semua program migrasi data dan menjalankannya dalam urutan yang benar. Bahkan proyek migrasi untuk perusahaan kecil dan menengah dapat mencakup ratusan program migrasi data. Tidak seorang pun seharusnya dilupakan. Pesanan juga tidak dapat diubah dengan mudah karena ketergantungan antara *object* bisnis. Ini adalah alasan mengapa tugas orkestrasi seharusnya otomatis. Meskipun ada otomatisasi, risiko tetap ada tidak semua program migrasi data dijalankan di Internet urutan yang benar.
- Program migrasi data dan komponen orkestrasi bergantung pada infrastruktur. Risiko infrastruktur lihat risiko yang terkait dengan infrastruktur teknis.

- Dimensi risiko mencerminkan ketidaklayakan di bawah tinggi beban. Ini bisa di lapisan jaringan, penyimpanan, atau bahkan lapisan basis data. Untuk memberikan contoh konkret, undo log bisa meluap karena migrasi program memiliki karakteristik berbeda dari biasanya aplikasi.
- Risiko gangguan muncul jika tidak hanya tim migrasi data yang aktif pada aplikasi sumber selama cut-over, tetapi juga pengguna, pengembang, atau kumpulan proses. Misalnya, jika satu pengguna mengakses table menyimpan semua pelanggan, ini dapat menghasilkan kunci meja. Kunci meja selama cut-over akan menghalangi migrasi data pelanggan, dan karenanya, meledak dengan mudah proyek migrasi data yang lengkap.
- Risiko parameterisasi aplikasi target adalah risiko yang berasal dari aplikasi target. Jika aplikasi target berubah, itu bisa menjadi tidak kompatibel dengan data program migrasi. Misalnya, jika aplikasi target parameterisasi diubah sedemikian rupa sehingga tidak diizinkan tambahkan pelanggan tanpa identifikasi yang tepat, ini mungkin mencerminkan peraturan hari ini. Namun, jika pelanggan dan akun ada di aplikasi sumber yang seharusnya pasti bermigrasi. Interaksi antara migrasi data dan parameterisasi aplikasi target dapat menyebabkan masalah. Sisa dari makalah ini berfokus pada risiko data proyek migrasi dapat memengaruhi secara langsung: risiko proyek.

## **2.7 Dimensi Manajemen Migrasi Data**

Di bagian ini kami menjelaskan solusi yang disarankan untuk mewujudkan proyek migrasi data. Selanjutnya, kami menentukan penggunaan yang diperlukan dari model pendekatan yang disajikan dalam makalah ini sebagai panduan untuk berhasil mengimplementasikan proyek migrasi data yang didukung oleh manajer proyek. Kami ingin menyajikan metode ini sebagai praktik terbaik. Gambar di bawah ini menggambarkan dimensi utama untuk proyek migrasi data yang berhasil. Singkatnya, kami mengklasifikasikan empat dimensi utama dari migrasi data:



**Gambar 2.4 Empat Dimensi Migrasi Data [7]**

### **2.7.1. Dimensi organisasi**

Apa saja batasan untuk sumber daya, garis waktu, penutupan dan kegiatan, dan departemen bisnis yang terlibat.

### **2.7.2. Standar prosedural**

Bagaimana prosedur yang diatur untuk memproses migrasi?

### **2.7.3. Dimensi bisnis**

Apa hasil yang diharapkan yang dibutuhkan oleh Bisnis Departemen? Di sini kita harus mempertimbangkan ruang lingkup fungsional, persyaratan pelaporan, dan sebagainya.

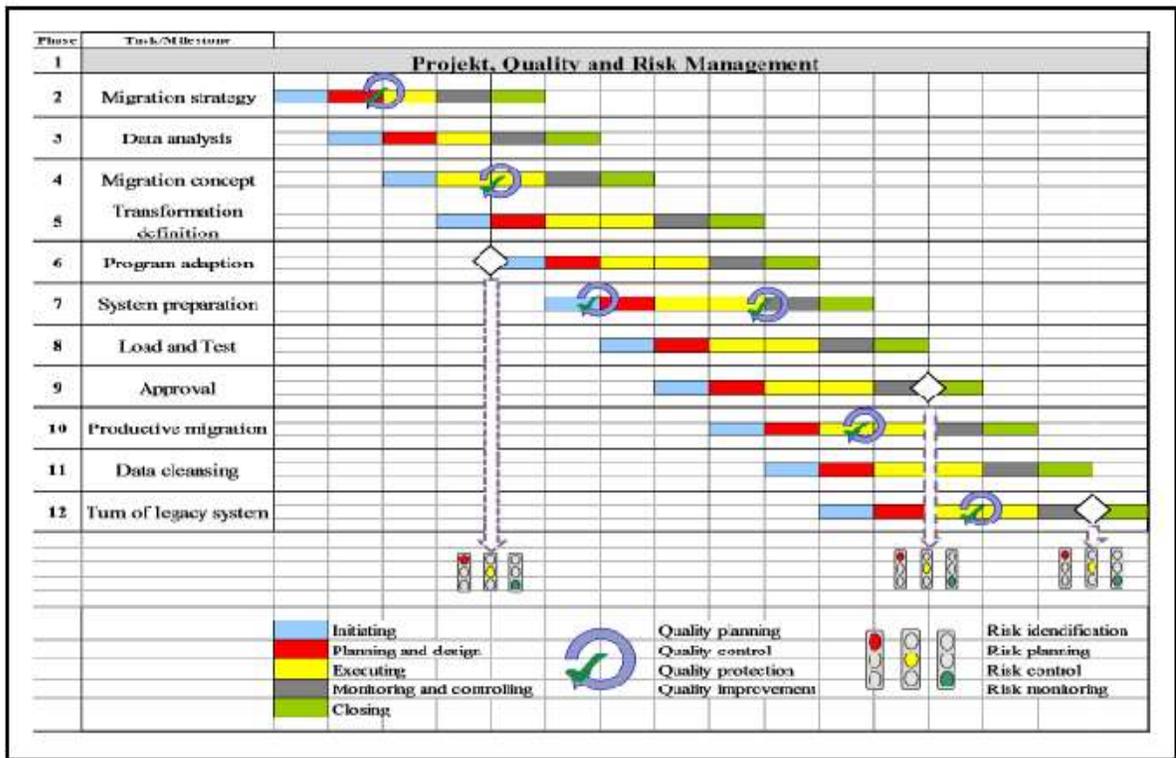
### **2.7.4 Dimensi Teknis**

Apa program atau format yang digunakan dalam migrasi? Dalam praktiknya kita harus fokus pada lanskap sistem yang berbeda, model pendekatan, dan sebagainya. Dalam makalah ini kami fokus pada aspek teknis dari proses migrasi data. Namun, kami juga mempertimbangkan dimensi organisasi dari proses manajemen migrasi data. Klasifikasi pada Gambar 2.4 membantu kita menunjukkan hanya aspek ini saja. Seperti yang dapat kita lihat di bagian berikut, artikel ini bertujuan untuk menyajikan model pendekatan untuk menghadapi proyek migrasi data. Di luar itu, bagian selanjutnya memberikan contoh detail dari dimensi teknis.

## **2.8 Pendekatan Migrasi Data**

Bagian ini membahas perincian proses migrasi data. Awalnya, kami memberikan definisi tentang pengertian migrasi data dengan merujuk pada literatur yang relevan dari bidang migrasi data. Migrasi data adalah proses teknis agregasi

dan / atau disagregasi entitas informasi dari sistem embedding untuk mencocokkan persyaratan pertukaran informasi termasuk penyesuaian format data sesuai kebutuhan. Jika organisasi tidak memiliki model pendekatan untuk kegiatan migrasi data mereka, bagaimana mereka bisa menjamin pelanggan migrasi data yang sukses? Kami membutuhkan pedoman dan karakteristik yang serius seperti pendekatan migrasi data yang dijelaskan di bawah ini. Pendekatan referensi lain untuk perencanaan proyek migrasi data serius ditunjukkan oleh. Harus dikatakan bahwa pakar migrasi yang intensif biaya dan pemeliharaan dengan harga tinggi untuk lingkungan sistem yang kompleks adalah kebutuhan dasar untuk keberhasilan penggunaan pendekatan migrasi Pendekatan migrasi adalah dua belas langkah berikut dalam ikhtisar:



Gambar 2.5 The Twelve Phases of Data Migration [7]

### A. Proyek, Kualitas dan Manajemen Risiko

Selama proses migrasi data yang terintegrasi (seperti dapat dilihat pada gambar 2.5), tugas manajer konversi data adalah untuk mengatur dan mengelola semua paket kerja berikut, yang sejalan dengan proyek yang unggul, kualitas dan manajemen risiko untuk migrasi data.

## B. Strategi Migrasi

Tantangan utama adalah menentukan strategi migrasi yang direkomendasikan yang memenuhi semua persyaratan pelanggan. Langkah ini berakhir dengan definisi cetak biru bisnis yang lebih terperinci. Cetak biru bisnis harus mencakup kegiatan-kegiatan berikut:

- Definisi kegiatan migrasi.
- Spesifikasi persyaratan fungsional.
- Kuantifikasi berbagai struktur *object* bisnis, prosedur, proses, dan *object*.
- Definisi data yang tidak boleh dimigrasi. langkah pertama dalam data migrasi adalah mengidentifikasi semua aplikasi dan data yang relevan yang digunakan, karena itu tidak akan cukup hanya dengan mengidentifikasi data.
- Definisi pengarsipan data historis yang tidak boleh dimigrasi.
- Ketergantungan pada sistem target dan sumber.

Hasil dari semua kegiatan di atas mengarah pada strategi migrasi tingkat tinggi. Kegiatan-kegiatan ini sebagian besar dapat berasal dari TOGAF ADM, tepatnya dari fase "peluang dan solusi" dan "perencanaan migrasi".

## C. Analisis Data

proses migrasi bisa gagal untuk mengatasi masalah mendasar kualitas data. Di bawah sudut pandang ini, langkah analisis data sangat penting untuk penerimaan penuh dari data yang dimigrasi dalam sistem target. Sasaran untuk migrasi data yang sukses bertepatan dengan persyaratan pelanggan. Ini hanya dapat dicapai dengan bekerja berdampingan dengan pengguna akhir bisnis. Masalah yang relevan dalam langkah analisis data dirangkum sebagai berikut:

- Pendekatan untuk membongkar data sumber dan analisis data sumber.
- Konstruksi infrastruktur migrasi (fokus pada alat migrasi) dan bekerja di luar spesifikasi teknis.
- Dokumentasi data historis (mis. Penggunaan dokumentasi sistem yang ada).
- Ada duplikat dalam sistem sumber dan dampaknya terhadap sistem target.
- Teknik *Reverse*.
- Skenario impor potensial (manual, semi-teknis, teknis).
- Penataan data (tidak terstruktur-sangat terstruktur),
- Menentukan format data dan sumber data.
- Identifikasi tempat penyimpanan informasi (struktur aplikasi 3 tingkat).

#### **D. Konsep Migrasi**

Konsep migrasi mengilustrasikan tampilan gambaran besar dari drive migrasi data. Ini termasuk masalah-masalah berikut:

- Definisi lingkungan teknis untuk pengembangan dan pengujian.
- Definisi alat dan program migrasi. Misalnya, walaupun lebih disukai untuk memiliki dukungan alat homogen, diperkirakan bahwa kita harus mempertahankan dukungan alat heterogen selama fase migrasi. Tetapi penting untuk mengetahui bahwa di satu sisi alat migrasi dapat sangat membantu, dan di sisi lain alat migrasi bukan solusi total.
- Definisi *gateway* untuk sistem lama untuk ekstraksi data.
- Definisi benda uji dan kriteria persetujuan.
- Tes fungsional diferensiasi, tes migrasi, tes integrasi.
- Definisi ketergantungan antara sistem lama dan sistem lain yang digunakan.
- Deskripsi siklus pengembangan sistem warisan dan ketergantungannya.

#### **E. Definisi Transformasi**

Tahap transformasi berkaitan dengan tugas restrukturisasi kegiatan migrasi. Dalam proses ini kita harus menyelesaikan beberapa tugas:

- Pemetaan data lama ke arsitektur target dan model data target.
- Definisi aturan pemetaan termasuk logika perhitungan untuk mengkonversi data lama dan Definisi untuk bukti konsep yang tepat.
- Konsolidasi data, pengayaan data dan format ulang atau pengkodean data.
- Data yang diperlukan atau opsional dalam struktur data target.
- Definisi aturan transformasi dan pendekatan operatif / deklaratif.
- Salah satu tantangan terbesar dalam fase migrasi ini adalah menemukan keahlian [2] di bidang bisnis baik sistem warisan maupun sistem target.

#### **F. Program Adaptasi**

Fase adaptasi program bertujuan untuk menyaring lingkungan pelanggan (mis. Logika pelanggan khusus, bidang pelanggan khusus, dan persyaratan logging khusus) untuk mencapai penggunaan ulang program yang ada sebanyak mungkin.

#### **G. Persiapan Sistem**

Tugas migrasi data berikut adalah perlunya mempersiapkan lingkungan sistem untuk migrasi:

- Implementasi kerangka kerja migrasi.

- Data produksi diekstraksi atau dibekukan dari data produksi.
- Salinan produksi (cadangan produksi) sistem warisan.
- *Object* disiapkan (pembekuan fungsional untuk sistem warisan).

## **H. Muat dan Uji**

Fase muat dan uji akan dijelaskan dengan masalah di bawah ini:

- Transfer data ke lingkungan pengujian fungsional dependensi.
- Tes kelengkapan, konsistensi, integritas; dan non-pengambilalihan.
- Validasi hasil bisnis.
- Penciptaan pendekatan untuk verifikasi dan pemeriksaan konsistensi.
- Pengujian pendekatan transformasi.
- Pengujian aplikasi menggunakan data yang dimigrasi.
- Tes integrasi dan uji coba.

## **I. Persetujuan**

Pada tahap ini bukti konsep harus ditinjau oleh semua pihak yang terlibat dalam migrasi data. Ini harus ditentukan jika kriteria kualitas cukup terpenuhi dan apakah aturan migrasi dinamis dapat diverifikasi. Unit persetujuan akan ditutup dengan tanda untuk migrasi produktif.

## **J. Migrasi Produktif**

Fase migrasi produktif mengandung tiga masalah utama: Pertama, transfer data ke sistem produksi dilakukan. Kedua, skenario mundur harus disediakan dan akhirnya prosedur eskalasi diberlakukan.

## **K. Pembersihan Data**

Selama langkah pembersihan data, pakar migrasi menghabiskan waktu untuk pengayaan data - misalnya adaptasi terhadap lingkungan fungsional yang berubah, untuk menambah keberpihakan dengan implementasi rilis. fase ini memberikan ahli konversi kesempatan terbaik untuk menghilangkan data yang mati dari sumber data.

## **L. Matikan Sistem Lawas**

Langkah terakhir adalah mematikan sistem legacy dan mewujudkan migrasi data, yang diakhiri dengan pemotongan dari sistem legacy. Dengan demikian, pakar konversi harus mengingat masalah berikut:

- Sistem *Freeze of legacy* untuk penggunaan di masa depan.
- Salinan akhir disiapkan.
- Sistem lama yang disediakan di "view" -mode dan pengarsipan data historis.