

**TUGAS AKHIR – IF184802**

**PENDEKATAN BERBASIS GRAF DALAM PENENTUAN KESESUAIAN ANTARA LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

**NAMA MAHASISWA**

NRP 5025201043

Dosen Pembimbing 1

**Ir. Adhatus Solichah Ahmadiyah, S.Kom, M.Sc**

NIP 1985082620150420002

Dosen Pembimbing 2

**Kelly Rossa Sungkono, S.Kom., M.Kom.**

NIP 1994201912088

**Program Studi S1 Teknik Informatika**

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

1



**TUGAS AKHIR – IF184802**

**PENDEKATAN BERBASIS GRAF DALAM PENENTUAN KESESUAIAN ANTARA LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

**Fahmi Muhazir**

NRP 5025201043

Dosen Pembimbing 1

**Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.**

NIP 198508262015042002

Dosen Pembimbing 2

**Kelly Rossa Sungkono, S.Kom., M.kom.**

NIP 1994201912088

# HALAMAN JUDUL

**Program Studi S1 Teknik Informatika**

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya

2024



**FINAL PROJECT – IF184802**

**PENDEKATAN BERBASIS GRAF DALAM PENENTUAN KESESUAIAN ANTARA LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

**Fahmi Muhazir**

NRP 5025201043

Advisor I

**Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.**

NIP 198408262015042002

Advisor 2

**Kelly Rossa Sungkono, S.Kom., M.Kom.**

NIP 1994201912088

**Study Program Bachelor of Informatics**

Department of Informatics

Faculty of Intelligent Electrical and Informatics Technology

Institut Teknologi Sepuluh Nopember

Surabaya

2024

# LEMBAR PENGESAHAN

**PENDEKATAN BERBASIS GRAF DALAM PENENTUAN KESESUAIAN ANTARA LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

Diajukan untuk memenuhi salah satu syarat

Memperoleh gelar Sarjana Komputer pada

Program Studi S-1 Teknik Informatika

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Oleh : **Fahmi Muhazir**

NRP. 5025201043

Disetujui oleh Tim Penguji Tugas Akhir:

|  |  |  |
| --- | --- | --- |
| 1. | Ir. Adhatus Solichah Admadiyah, S.Kom., M.Sc | Pembimbing |
| 2. | Kelly Rossa Sungkono, S.Kom, M.Kom. | Ko-pembimbing |
| 3. | Nama dan gelar penguji | Penguji |
| 4. | Nama dan gelar penguji | Penguji |
| 5. | Nama dan gelar penguji | Penguji |

**SURABAYA**

**Maret, 2024**

**APPROVAL SHEET**

**PENDEKATAN BERBASIS GRAF DALAM PENENTUAN KESESUAIAN ANTARA LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

Submitted to fulfill one of the requirements

for obtaining a degree Bachelor of Computer at

Undergraduate Study Program of S-1 Informatics

Department of Informatics

Faculty of Intelligent Electrical Engineering and Information of Technology

Institut Teknologi Sepuluh Nopember

By: **Fahmi Muhazir**

NRP.5025201043

Approved by Final Project Examiner Team:

|  |  |  |
| --- | --- | --- |
| 1. | Name of Advisor and academic title | Advisor |
| 2. | Name of Co-Advisor/Examiner and academic title | Co-Advisor |
| 3. | Name of Examiner and academic title | Examiner |
| 4. | Name of Examiner and academic title | Examiner |
| 5. | Name of Examiner and academic title | Examiner |

**SURABAYA**

**Maret, 2024**

# PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Departemen :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dosen Pembimbing / NIP :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Judul Tugas Akhir” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari dietemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima saksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

|  |  |
| --- | --- |
|  | Surabaya, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Mengetahui |  |
| Dosen Pembimbing | Mahasiswa |
|  |  |
| (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) | (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) |
| NIP. | NRP. |

**STATEMENT OF ORIGINALITY**

The undersigned below:

Name of student / NRP :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Advisor / NIP :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Hereby declare that Final Project with the title of “Title of Final Project” is the result of my own work, is original, and is written by following the rules of scientific writing.

If in the future there is a discrepancy with this statement, then I am willing to accept sanctions

in accordance with the provisions that apply at Institut Teknologi Sepuluh Nopember.

|  |  |
| --- | --- |
|  | Surabaya, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Acknowledge |  |
| Advisor | Student |
|  |  |
| (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) | (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) |
| NIP. | NRP. |

**PENDEKATAN BERBASIS GRAPH DALAM PENENTUAN KESESUAIAN ANTARA** **LAPISAN APLIKASI DAN TEKNOLOGI ARCHIMATE MENGGUNAKAN NEO4J**

**Nama Mahasiswa / NRP : Fahmi Muhazir / 5025201043**

**Departemen : Teknik Informatika FTEIC - ITS**

**Dosen Pembimbing 1 : Ir. Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.**

**Dosen Pembimbing 2 : Kelly Rossa Sungkono, S.Kom., M.Kom.**

# ABSTRAK

**Abstrak**

ArchiMate adalah aplikasi pemodelan *enterprise architecture* yang terintegrasi dengan kerangka kerja TOGAF. Pada ArchiMate, layer Aplikasi digunakan untuk menggambarkan struktur dan fungsi aplikasi perusahaan, sementara layer Teknologi berisi menggambarkan dari infrastruktur teknologi yang mendukung aplikasi tersebut. Design yang baik menekankan bahwa layer Sistem Informasi (SI) dan Teknologi Informasi (TI) harus saling sesuai untuk mencapai integrasi yang optimal. Namun, hingga saat ini, belum ada alat bantu yang dapat secara efektif mengukur kesesuaian antara layer Aplikasi dan Teknologi Informasi. Oleh karena itu pada tugas akhir ini akan dibangun pengukuran kesesuaian antara layer Aplikasi dan Teknologi Informasi. Knowledge Graph dengan Neo4J akan digunakan untuk menyajikan hubungan dan ketergantungan antara elemen-elemen pada kedua layer. Dengan adanya tugas akhir ini diharapkan dapat mengukur kesesuaian antar layer, sehingga memudahkan evaluasi model arsitektur dengan hasil akurasi persentase kesesuaian layer Aplikasi terhadap layer Teknologi Informasi.

**Kata kunci: ArchiMate, Enterprise Architecture, Knowledge Graph, Layer Aplikasi, Layer Teknologi, TOGAF.**

**GRAPH-BASED APPROACH IN DETERMINING THE COMPATIBILITY BETWEEN APPLICATION LAYERS AND ARCHIMATE TECHNOLOGY USING NEO4J**

**Student Name / NRP : Fahmi Muhazir / 5025201043**

**Department : Informatic Engineering FTEIC - ITS**

**Advisor : Adhatus Solichah Ahmadiyah, S.Kom., M.Sc.**

**Co-Advisor : Kelly Rossa Sungkono, S.Kom., M.Kom.**

**Abstract**

ArchiMate is an enterprise architecture modeling application that is integrated with the TOGAF framework. In ArchiMate, the Application layer is used to describe the structure and function of enterprise applications, while the Technology layer contains a description of the technology infrastructure that supports these applications. A good design emphasizes that the Information Systems (IS) and Information Technology (IT) layers must fit together to achieve optimal integration. However, until now, there is no tool that can effectively measure the compatibility between the Application and Information Technology layers. Therefore, in this final project, a measurement of compatibility between the Application and Information Technology layers will be built. Knowledge Graph with Neo4J will be used to present the relationships and dependencies between elements in both layers. With this final project, it is expected to be able to measure the suitability between layers, making it easier to evaluate the architecture model with the results of the accuracy of the percentage of suitability of the Application layer to the Information Technology layer.

**Kata kunci: ArchiMate, Enterprise Architecture, Knowledge Graph, Layer Aplikasi, Layer Teknologi, TOGAF.**

# KATA PENGANTAR

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# DAFTAR ISI

[HALAMAN JUDUL i](#_Toc107480714)

[LEMBAR PENGESAHAN iii](#_Toc107480715)

[PERNYATAAN ORISINALITAS v](#_Toc107480716)

[ABSTRAK vii](#_Toc107480717)

[KATA PENGANTAR ix](#_Toc107480718)

[DAFTAR ISI x](#_Toc107480719)

[DAFTAR GAMBAR/GRAFIK/DIAGRAM xi](#_Toc107480720)

[DAFTAR TABEL xii](#_Toc107480721)

[DAFTAR SIMBOL (jika ada) xiii](#_Toc107480722)

[DAFTAR SINGKATAN (jika ada) xiv](#_Toc107480723)

[BAB I PENDAHULUAN 1](#_Toc107480724)

[1.1 Latar belakang 1](#_Toc107480725)

[1.2 Rumusan Permasalahan 1](#_Toc107480726)

[1.3 Batasan Masalah 1](#_Toc107480727)

[1.4 Tujuan 1](#_Toc107480728)

[1.5 Manfaat 1](#_Toc107480729)

[BAB II TINJAUAN PUSTAKA 2](#_Toc107480730)

[2.1 Penelitian Terkait 2](#_Toc107480731)

[2.2 Dasar Teori 2](#_Toc107480732)

[BAB III METODOLOGI 3](#_Toc107480733)

[3.1 Metode yang dirancang 3](#_Toc107480734)

[3.2 Peralatan pendukung 3](#_Toc107480735)

[3.3 Implementasi 3](#_Toc107480736)

[BAB IV HASIL DAN PEMBAHASAN 4](#_Toc107480737)

[4.1 Hasil eksperimen 4](#_Toc107480738)

[4.2 Pembahasan/Diskusi 4](#_Toc107480739)

[BAB V KESIMPULAN DAN SARAN 5](#_Toc107480740)

[5.1 Kesimpulan 5](#_Toc107480741)

[5.2 Saran (jika dianggap perlu) 5](#_Toc107480742)

[DAFTAR PUSTAKA 6](#_Toc107480743)

[LAMPIRAN-LAMPIRAN ATAU APPENDIKS (jika ada) 7](#_Toc107480744)

[BIODATA PENULIS 8](#_Toc107480745)

# DAFTAR GAMBAR/GRAFIK/DIAGRAM

# DAFTAR TABEL

# DAFTAR SIMBOL (jika ada)

# DAFTAR SINGKATAN (jika ada)

# BAB I PENDAHULUAN

## Latar belakang

Aplikasi telah menjadi komponen integral dalam operasi organisasi modern. Kemajuan teknologi dan informasi (TI) telah memungkinkan organisasi untuk meningkatkan efisiensi, produktivitas dan daya saingnya. Dalam usaha untuk mencapai tujuan tersebut, perlu ada perencanaan, perancangan dan pengelolaan Aplikasi yang baik. Dalam konteks ini adalah pendekatan *Enterprise Architecture* menjadi komponen utama untuk mencapai tujuan dari bisnis.Pada pendekatan *Enterprise Architecture* diperlukan Tata Kelola TI(IT Governance) yang menjadi aspek penting dalam manajemen Teknologi Informasi, tujuannya adalah memastikan bahwa TI dikelola dengan baik sesuai dengan tujuan organisasi, risiko, dan kepatuhan. Tata kelola TI membantu organisasi dalam mengambil keputusan yang tepat terkait dengan investasi, keamanan, dan pengelolaan sumber daya TI.

TOGAF (*The Open Group Architecture Framework*) adalah sebuah kerangka kerja arsitektur perusahaan yang telah diadopsi secara luas oleh banyak organisasi di seluruh dunia. TOGAF menyediakan panduan dan metode terstruktur untuk merencanakan, merancang, dan mengelola arsitektur perusahaan. Tujuannya adalah untuk membantu organisasi atau Perusahaan dalam mencapai keselarasan antara strategi bisnis, proses bisnis, dan Teknologi Informasi. Kemudian, ArchiMate adalah bahasa pemodelan visual yang dikembangkan oleh The Open Group untuk mendukung pemodelan, analisis, dan dokumentasi arsitektur enterprise. ArchiMate memberikan notasi dan konsep-konsep untuk menggambarkan berbagai aspek dari arsitektur enterprise, termasuk struktur aplikasi, proses bisnis, dan infrastruktur teknologi juga teradpat 3 lapisan/*layer* utama pada archimate yaitu lapisan bisnis, aplikasi dan teknologi.

Layer Aplikasi dalam ArchiMate mencakup aspek-aspek terkait dengan struktur dan fungsi aplikasi perusahaan, sedangkan layer Teknologi menggambarkan infrastruktur teknologi yang menjadi pendukung bagi aplikasi tersebut. Desain yang optimal memerlukan kesesuaian yang baik antara layer Sistem Informasi (SI) dan Teknologi Informasi (TI). Meskipun demikian, belum ada alat bantu yang efektif untuk mengukur sejauh mana kesesuaian ini tercapai. Oleh karena itu, tugas akhir ini diinisiasi dengan tujuan untuk membangun suatu metode pengukuran kesesuaian antara layer Aplikasi dan Teknologi Informasi dalam konteks ArchiMate.

Penggunaan Knowledge Graph dengan Neo4J sebagai alat bantu diharapkan dapat menyajikan hubungan dan ketergantungan antara elemen-elemen pada kedua layer secara lebih terstruktur. Kesesuaian antar layer ini sangat penting untuk mencapai integrasi yang optimal, efisiensi, dan keberlanjutan sistem informasi di tingkat enterprise. Dengan adanya kontribusi dari tugas akhir ini, diharapkan dapat memberikan pemahaman yang lebih baik tentang hubungan antara layer Aplikasi dan Teknologi, yang pada akhirnya dapat meningkatkan kualitas dan efektivitas arsitektur enterprise secara keseluruhan.

## Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana mengidentifikasi notasi dan relasi pada *layer* Aplikasi ArchiMate ?
2. Bagaimana mengidentifikasi notasi dan relasi pada *layer* Teknologi Informasi ArchiMate
3. Bagaimana mekanisme penentuan kesesuaian antara fase arsitektur Aplikasi dengan arsitektur teknologi pada TOGAF dengan menggunakan kakas bantu ArchiMate ?

## Batasan Masalah

Batasan masalah yang dibahas dalam Tugas Akhir ini adalah sebagai berikut.

1. Kerangka kerja yang digunakan sebagai acuan dalam menganalisis arsitektur yang dievaluasi adalah TOGAF 9 edisi 3.
2. Versi ArchiMate yang digunakan adalah versi 5.2.0.
3. Basis Data Graf yang digunakan sebagai media visualisasi dan analisis kesesuaian pemodelan adalah Neo4J versi 1.5.9.

## Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. untuk mengetahui bagaimana korelasi antara *layer* arsitektur Aplikasi dan Teknologi Informasi,
2. untuk mengetahui bagaimana cara untuk menyelaraskan hubungan antara *layer* arsitektur Aplikasi dan arsitektur Teknologi Informasi.

## Manfaat

Melalui Tugas Akhir diharapkan dapat mengidentifikasi kebutuhan dan peluang untuk mengembangkan kakas bantu yang dapat mendukung arsitek dan profesional TI dalam menyelaraskan *layer* arsitektur Aplikasi dan Teknologi Informasi yang dirancang.

# BAB II TINJAUAN PUSTAKA

## Penelitian Terkait

Dalam penyusunan tugas akhir ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian-penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada skripsi ini. Seperti dalam artikel Naranjo *et al* pada tahun 2015 yang membahas pendekatan berbasis graf untuk analisis arsitektur perusahaan, yang dikenal sebagai PRIMROSe. Pendekatan ini dirancang untuk memberikan pemahaman yang lebih dalam tentang model perusahaan [1]. Kemudian pada artikel Muhamed Smajevic *et al* pada tahun 2021yang membahas pengembangan kerangka kerja generik untuk mentransformasi model Arsitektur Enterprise (EA) menjadi struktur graf guna analisis otomatis. Implementasi platform awan Analisis Arsitektur Enterprise berbasis Graf (eGEAA) juga dibahas. Artikel menyoroti manfaat analisis graf dalam pengambilan keputusan bagi arsitek perusahaan, dan kerangka kerja ini diharapkan membantu mengatasi kompleksitas pada model EA yang semakin berkembang [2].

## Dasar Teori

## Enterprise Architecture

*Enterprise Architecture* merupakan sebuah struktur, kebiasaan atau perilaku, alur informasi, termasuk personil dan subdivisi dari sebuah perusahaan atau organisasi, dalam suatu proses bisnis untuk mencapai tujuan dari perusahaan sejalan dengan proses bisnis yang telah ditentukan. *Enterprise Architecture* tidak selalu berkaitan dengan Aplikasi, namun dengan seiring berkembangnya teknologi, arus infromasi digital menjadi semakin tinggi untuk mempermudah proses bisnis. Karena hal tersebut Enterprise Architecture menjadi penting untuk merancang proses bisnis yang sesuai dan efektif untuk mencapai tujuan dari perusahaan atau organisasi berdasarkan proses bisnis yang telah ditentukan [3].

## TOGAF

TOGAF merupakan standar kerangka kerja yang menyediakan metode dan alat untuk membantu mengadopsi, membangun dan pemeliharaan dari *Enterprise Architecture.* Tujuanya adalah untuk menyelaraskan tujuan TI selaras dengan tujuan dari perusahaan atau organisasi sehingga saling mendukung dan dapat mencapai tujuan bisnis [4]. Pada TOGAF terdapat metode inti yang disebut dengan ADM (*Architecture Development Method*) yang merupakan metode iteratif yang berkelanjutan, sehingga perusahaan dapat mengontrol perkembangan perusahaan berdasarkan tujuan dan peluang bisnis. Terdapat 8 fase dalam ADM (seperti ditampilkan pada Gambar 2.1), ke 8 fase tersebut diinisiasi oleh fase Preliminary yang mendefinisikan kapabilitas, penyesuaian kerangka TOGAF dan prinsip arsitektur. Berikut merupakan 8 fase setelah preliminary [5]:

A diagram of a diagram

Description automatically generated

Gambar 2.1 Fase-fase pada TOGAF ADM

1. *Architecture Vision*

*Architecture Vision* adalah fase yang menentukan *scope,* mengidentifikasi stakeholders, menentukan ruang lingkup dan batasan. Berdasarkan aspek yang telah ditentukan, akan terbentuk sebuah *Architecture Vision,* validasi konteks bisnis dan persetujuan dari desain arsitektur.

Dalam membuat *Architecture Vision,* Arsitek dapat membaca dokumentasi perusahaan untuk memahami dan memverifikasi tujuan, visi dan strategi dari perusahaan. *Architecture Vision* ini menjadi *outline* untuk menentukan fase-fase selanjutnya dan harus tertuang dalam sebuah dokumentasi (*Architecture Work*), tentunya dokumentasi tersebut harus divalidasi oleh *stakeholder* terkait dengan menggunakan konsensus secara legal.

1. *Business Architecture*

*Business Architecture* adalah fase yang menjelaskan tentang produk atau strategi servis, struktur organisasi, fungsi, proses, informasi dan aspek lingkungan perusahaan. Dengan kata lain, Business Architecture merupakan sebuah bentuk visualisasi nilai bisnis dari *Architecture Work* kepada *stakeholder.* Ini mengacu pada return investment dari keterlibatan *stakeholder*. Terdapat 2 tujuan utama dalam fase ini:

1. Target bisnis arsitektur, ini menjelaskan tentang bagaimana *action* Perusahaan untuk mencapai tujuan bisnisnya. Hal ini akan melibatkan respon pada faktor pendorong yang telah ditetapkan pada *Architecture Vision.*
2. Identifikasi rencana arsitektur, dilakukan untuk melihat kesenjangan pada kerangka dasar bisnis (*Baseline Business Architecture*)dan target bisnis arsitektur (Target Business Architecture).
3. *Information System Architecture*

*Information System Architecture* adalah fase yang menjelaskan mengembangan sistem di mana perencanaan, desain, dan konseptualisasi arsitektur Aplikasi dilakukan. Pengembangan pada fase ini dilakukan pada 2 bagian yaitu data dan arsitektur aplikasi. Terdapat 2 tujuan dalam fase ini:

* 1. Pengembangan arsitektur Aplikasi adalah proses untuk merinci bagaimana Arsitektur informasi pada Perusahaan akan mendukung arsitektur bisnis dan arsitektur teknologi dalam kebutuhan bisnis perusahaan
  2. Identifikasi Rencana Arsitektur dilakukan untuk melihat kesenjangan pada kerangka dasar bisnis terhadap target arsitektur Aplikasi.

1. *Technology Architecture*

*Technology Architecture* adalah fase yang menjelaskan tentang pengembangan arsitektur teknologi yang tujuannya adalah merencanakan, mengelola dan mengintegrasikan arsitektur teknologi untuk mendukung kebutuhan bisnis dan strategi dari perusahaan. Terdapat 2 tujuan pada fase ini:

1. Pengembangan target arsitektur teknologi adalah bagaimana komponen dapat terorganisir dan mendukung aplikasi juga data baik secara logis maupun fisik. Ini mencakup *hardware, software, platform,* jaringan dan infrastruktur terkait
   1. Identifikasi Rencana Arsitektur dilakukan untuk melihat kesenjangan pada kerangka dasar bisnis terhadap target arsitektur teknologi.
2. *Opportunities and Solutions*

*Opportunities and Solutions* adalah proses identifikasi peluang, alternatif, strategi dan implementasi. Fase ini penting untuk membantu perusahaan untuk tetap relevan dengan segala perubahan yang akan dihadapi nanti. Terdapat 3 tujuan pada fase ini:

1. Membuat versi lengkap dari *roadmap* arsitektur, berdasarkan pada Analisis kesenjangan dan kandidat roadmap arsitektur pada fase B, C dan D.
2. Menganalisis kembali apakah perlu dilakukan pendekatan secara incremental, analisis dan identifikasi juga arsitektur transisi yang akan memberikan nilai bisnis.
3. *Migration Planning*

*Migration Planning* adalah fase yang menjelaskan rencana *baseline* untuk mencapai target dari arsitektur perusahaan yang dibuat:

1. Finalisissi Rencana Arsitektur untuk mendukung fase selanjutnya.
2. Memastikan rencana implementasi dan migration sudah selaras dengan pendekatan yang dilakukan Perusahaan untuk mengelola perubahan pada Perusahaan.
3. Memastikan nilai bisnis dan biaya yang ada juga transisi arsitektur dipahami dengan baik oleh pihak *stakeholder* terkait
4. *Implementation Governance*

*Implementation Governance* adalah fase yang berfokus pada pengawasan dan pengelolaan implementasi proyek-proyek yang terkait dengan perubahan arsitektur yang telah direncanakan dalam fase *Planning and Migration.* Tujuan dari fase ini adalah memastikan bahwa perubahan arsitektur dilaksanakan sesuai dengan rencana yang telah disetujui dan mencapai tujuan arsitektur yang telah ditetapkan. Terdapat 2 tujuan pada fase ini :

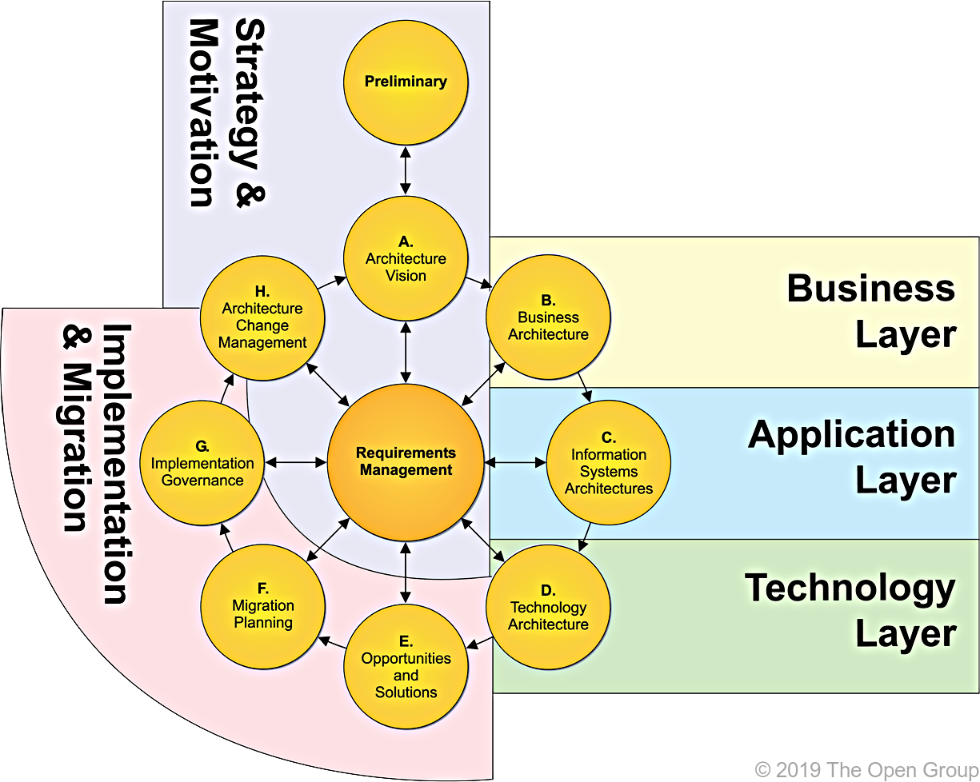
1. Memastikan kesesuaian untuk implementasi projek dengan target arsitektur yang telah ditetapkan
2. Melakukan fungsi tata kelola arsitektur dalam tiap pengembangan solusi sebagai respon dari setiap permintaan yang muncul.
3. *Architecture Change Management*

*Architecture Change Management* adalah fase yang terjadi atau berjalan sepanjang siklus dari ADM. Fase ini bertujuan untuk monitoring dan manajemen perubahan yang terjadi untuk memastikan kesesuaian arsitektur dengan kebutuhan dari bisnis secara optimal. Terdapat 3 tujuan dalam fase ini:

1. Memastikan bahwa *lifecycle* dari arsitektur terkelola dengan baik
2. Memastikan bahwa *framework* tata kelola dilakukan.
3. Memastikan bahwa arsitektur perusahaan sesuai atau memenuhi kebutuhan dari Perusahaan.

## ArchiMate

ArchiMate merupakan bahasa pemodelan untuk *Enterprise Architecture* yang dapat mendeskripsikan, menganalisis dan memodelkan arsitektur bisnis di tiap domain (*business, application and technology*) [6]. ArchiMate dan TOGAF ADM saling berkaitan, TOGAF sebagai sebuah panduan umum dan penggambaran proses siklus hidup dari *Enterprise Architecture*, sementara Archimate menyediakan notasi dan konsep-konsep untuk menggambarkan berbagai aspek dari arsitektur enterprise, termasuk struktur aplikasi, proses bisnis, dan infrastruktur teknologi. Seperti pada Gambar 2.3 yang menggambarkan korelasi antara TOGAF dan Archimate, diaman terdapat kesesuaian TOGAF pada fase arsitektur Bisnis (B), arsitektur Sistem Informasi (C) dan arsitektur Teknologi (D) dengan lapisan Bisnis, Aplikasi dan teknologi pada ArchiMate.



Gambar 2.2 Korespondensi antara ArchiMate dan TOGAF ADM[7]

Struktur utama dari elemen perilaku/*behavior* dan struktur/structure dalam ArchiMate dapat dilihat dalam fragmen metamodel pada Gambar 2.3. Gambar tersebut memberikan definisi umum untuk elemen-elemen tersebut tanpa memandang lapisan tertentu. Penting untuk dicatat bahwa sebagian besar elemen (kotak putih) dalam gambar tersebut bersifat abstrak dalam metamodel. Artinya, elemen-elemen tersebut tidak langsung diwujudkan dalam model, melainkan hanya berperan dalam menyusun kerangka kerja metamodel itu sendiri[7].

A diagram of elements with text

Description automatically generated

Gambar 2.3 Hierarki dari Behavior dan Structure Elements[7]

Fragmen metamodel pada Gambar 2.3 terdiri dari dua jenis elemen utama, yaitu elemen struktur ("kata benda") dan elemen perilaku ("kata kerja"). Elemen struktur dapat dibagi lebih lanjut menjadi elemen struktur aktif dan elemen struktur pasif, dengan elemen struktur aktif terbagi menjadi elemen struktur aktif eksternal (atau antarmuka) dan elemen struktur aktif internal. Di sisi lain, elemen perilaku dapat dikelompokkan menjadi elemen perilaku internal, elemen perilaku eksternal (atau layanan), dan peristiwa. Pengelompokan ini mencerminkan inspirasi dari *natural language*, di mana kalimat memiliki struktur subjek (struktur aktif), kata kerja (perilaku), dan objek (struktur pasif)[7]. Sebagai gambaran umumnya sepertu pada tabel 2.1.

Tabel 2.2.1. Structure dan Behavior Element [7]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Element | Specializations | Definition | | Notation | |
| Active Structure | | | | | | |
| Internal active structure element | | | Represents an entity that is capable of performing behavior. | | A white sign with black text  Description automatically generated | |
|  | Collaboration | Represents an aggregate of two or more internal active structure elements, working together to perform some collective behavior. | | A white and black text  Description automatically generated with medium confidence | |
| Interface (external active structure element) | | | Represents a point of access where one or more services are exposed to the environment. | | A white square with black text  Description automatically generated | |
| Behavior | | | | | | |
| Internal behavior element | | | Represents a unit of activity that can be performed by one or more active structure elements. | |  | |
|  | Process | Represents a sequence of behaviors that achieves a specific result. | | A close-up of a card  Description automatically generated | |
|  | Function | Represents a collection of behavior based on specific criteria, such as required resources, competencies, or location. | |  | |
|  | Interaction | Represents a unit of collective behavior that must be performed by two or more internal active structure elements, either assigned directly or aggregated in a collaboration. | |  | |
| Service (external behavior element) | | | Represents an explicitly defined exposed behavior. | | A white rectangular object with a black border  Description automatically generated | |
| Event | | | Represents a state change. | | A white rectangular object with a black corner  Description automatically generated | |
| Passive Structure | | | | | | |
| Passive structure element | | | Represents an element on which behavior is performed. | | A close up of a sign  Description automatically generated | |

Seperti pada Gambar 2.2, Archimate memiliki tiga lapisan/*layer* utama yaitu *Business Layer*, *Application Layer* dan *Technology Layer*. *Business Layer* dideskripsikan dengan *business element* dengan relasinya kepada domain arsitektur. Elemen pada business *layer* diinterpretasikan ke dalam 6 bentuk yaitu *business objective, product, business process/function/interaction, business actor/role, business service* dan *location*[7]*.*

*Application Layer* (Aplikasi)menggambarkan struktur, perilaku, dan interaksi aplikasi perusahaan.. Pada Lapisan Aplikasi terdapat 9 elemen, seperti pada Tabel 2*.2* lapisan ini mendukung lapisan sebelumnya yaitu Lapisan Bisnis[7]*.*

Tabel 2.2.2. Elemen pada Application Layer [7]

|  |  |  |
| --- | --- | --- |
| Element | Definition | Notation |
| Application component | Represents an encapsulation of application functionality aligned to implementation structure, which is modular and replaceable. |  |
| Application collaboration | Represents an aggregate of two or more application internal active structure elements that work together to perform collective application behavior. |  |
| Application interface | Represents a point of access where application services are made available to a user, another application component, or a node. |  |
| Application function | Represents automated behavior that can be performed by an application component. |  |
| Application interaction | Represents a unit of collective application behavior performed by (a collaboration of) two or more application components. |  |
| Application process | Represents a sequence of application behaviors that achieves a specific result. |  |
| Application event | Represents an application state change. |  |
| Application service | Represents an explicitly defined exposed application behavior. |  |
| Data object | Represents data structured for automated processing. |  |

Sebagai gambaran implementasi dari lapisan Aplikasi, saya mengambil contoh dari studi kasus ArchiSurance yang dibuat oleh The Open Group. Studi kasus ini mendeskripsikan secara umum bagaimaan arsitektur dari perusahaan, juga mengilustrasikan bagaimana penggunaan bahasa pemodelan ArchiMate berdasarkan TOGAF *Framework.* Seperti pada Gambar 2.4 yang menjelaskan tentang hubungan aliran data pada tiap komponen.

A diagram of a crm system

Description automatically generated

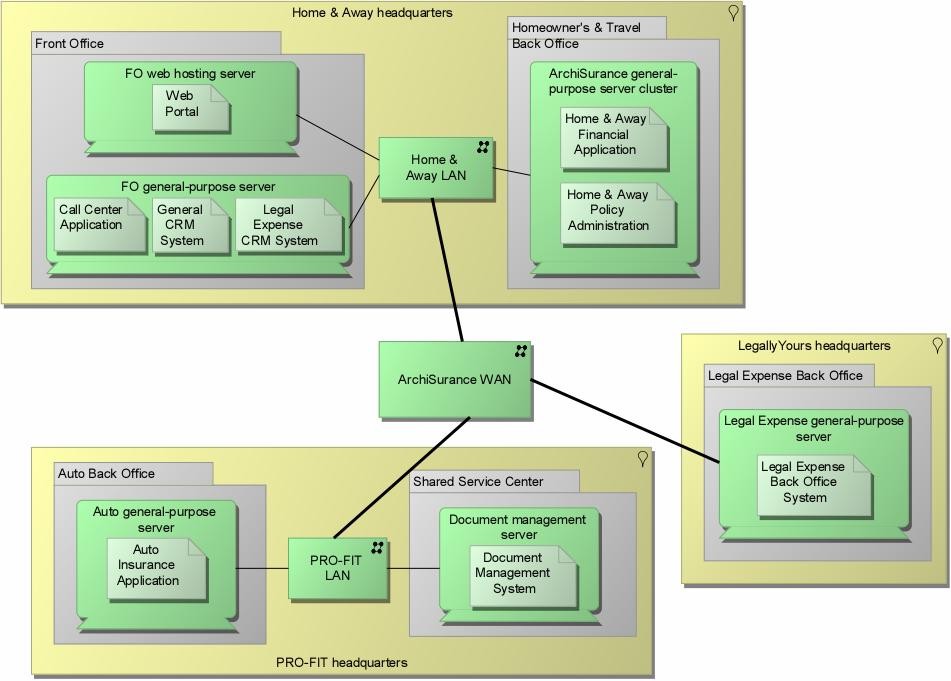
Gambar 2.4 Data flow dalam lapisan aplikasi[8]

*Technology or Infrastructure Layer* (Lapisan Teknologi) akan menggambarkanelemen dari infrastruktur perangkat lunak dan perangkat keras yang mendukung Lapisan Aplikasi, seperti perangkat fisik/*hardware*, jaringan/*network*, atau perangkat lunak sistem/system software. Terdapat 13 elemen pada lapisan teknologi seperti pada Tabel 2.3*.*

Tabel 2.2.3. Elemen pada Technology Layer[7]

|  |  |  |
| --- | --- | --- |
| Element | Definition | Notation |
| Node | Represents a computational or physical resource that hosts, manipulates, or interacts with other computational or physical resources. |  |
| Device | Represents a physical IT resource upon which system software and artifacts may be stored or deployed for execution. |  |
| System software | Represents software that provides or contributes to an environment for storing, executing, and using software or data deployed within it. |  |
| Technology collaboration | Represents an aggregate of two or more technology internal active structure elements that work together to perform collective technology behavior. |  |
| Technology interface | Represents a point of access where technology services offered by a node can be accessed. |  |
| Path | Represents a link between two or more nodes, through which these nodes can exchange data, energy, or material. |  |
| Communication network | Represents a set of structures that connects nodes for transmission, routing, and reception of data. |  |
| Technology function | Represents a collection of technology behavior that can be performed by a node. |  |
| Technology process | Represents a sequence of technology behaviors that achieves a specific result. |  |
| Technology interaction | Represents a unit of collective technology behavior performed by (a collaboration of) two or more nodes. |  |
| Technology event | Represents a technology state change. |  |
| Technology service | Represents an explicitly defined exposed technology behavior. |  |
| Artifact | Represents a piece of data that is used or produced in a software development process, or by deployment and operation of an IT system. |  |

Pada Gambar 2.5 menunjukkan komponen infrastruktur utama pada studi kasus ArchiSurance dan jaringan yang menghubungkan berbagai perangkat juga artefak (aplikasi) yang digunakan.

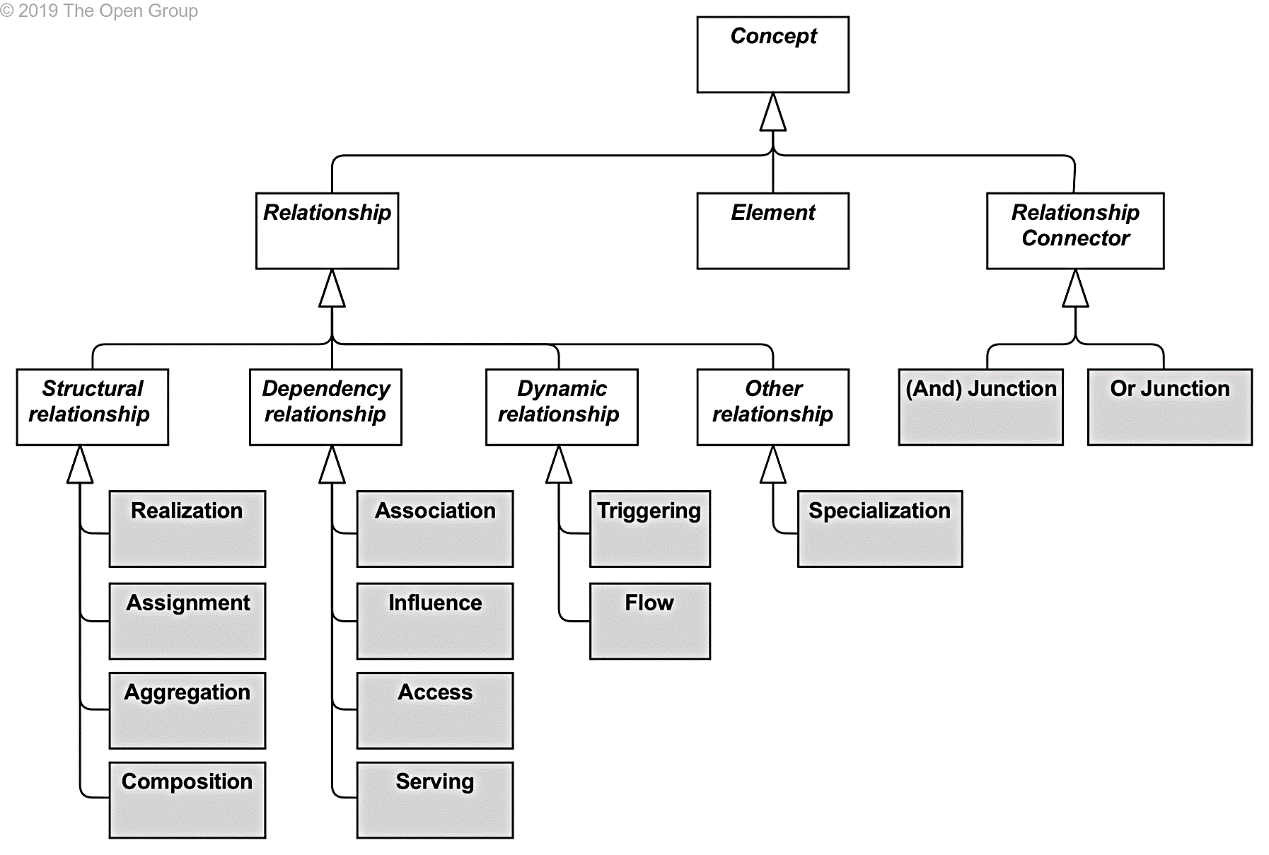


Gambar 2.5 Gambaran umum infrastruktur[8]

Bahasa ArchiMate mendefinisikan sekumpulan relasi/*relationship* generik utama yang masing-masing dapat menghubungkan jenis konsep tertentu dari awal hingga akhir. Umumnya, konsep-konsep ini berupa elemen, tetapi ada juga beberapa kasus di mana mereka bisa berupa hubungan lain. Sebagian besar dari hubungan-hubungan ini "overloaded," yang berarti makna pastinya berubah tergantung pada jenis konsep asal dan tujuan yang dihubungkan oleh hubungan tersebut.

Relasi pada ArchiMate di klasifikasina sebagai berikut (Gambar 2.2):

* Structural relationships, yang memodelkan konstruksi atau komposisi statis dari konsep-konsep yang sama atau berbeda jenis.
* Dependency relationships, yang memodelkan bagaimana elemen-elemen digunakan untuk mendukung elemen-elemen lainnya.
* Dynamic relationships, yang digunakan untuk memodelkan ketergantungan perilaku antara elemen-elemen.
* Other relationships, yang tidak termasuk dalam salah satu dari kategori di atas.



Gambar 2.6 Gambaran umum relasi pada ArchiMate[7]

## Neo4J

Neo4j adalah sistem manajemen basis data grafis (GDMBS) NoSQL berbasis Java yang menonjolkan skabilitas tinggi dan transaksi ACID. Platform ini menggunakan Knowledge Graphs untuk menyimpan data dan menyediakan Cypher Query Language (CQL) yang deklaratif, memungkinkan eksekusi query secara ekspresif dan langsung pada data grafik. CQL memudahkan interaksi intuitif dengan struktur grafik dan visualisasi graf, memfasilitasi pengambilan informasi yang efisien dari Knowledge Graphs dalam basis data Neo4J[9].

A screenshot of a computer

Description automatically generated

Gambar 2.7 Graph pada Sandbox Neo4J Projek "Recomendation"[10]

# BAB III METODOLOGI

Pada bab ini akan membahas analisis dan pendefinisan *rules* atau aturan yang akan digunakan untuk melakukan *alignment* element pada arsitektur sistem informasi tehadap teknologi Gambar 3.1. Setelah mendapatkan *rules* akan dilakukan uji coba atau implementasi terhadap salah satu studi kasus dari Archimate yaitu ArchIsurance (gambar 3.2) dan beberapa hasil tugas akhir kelas TKTI Teknik Informatika ITS tahun 2023. Adapun detail jelas dari proses yang akan dilakukan dapat dilihat pada gambar 3.1 dan gambar 3.2.

A black background with white rectangles

Description automatically generated

Gambar 3.1 Proses analisis dan pembuatan rules

A black background with white rectangles

Description automatically generated

Gambar 3.2 Implementasi rules dengan Neo4J untuk mendapatkan persentase kesesuaian

## Analisis

## Analisa Bentuk Relasi

Archimate menggunakan beberapa bentuk atau jenis ralasi seperti pada **Gambar 2.6**. Tiap relasi mempunyai fungsi nya masing-masing, namun terlepas dari hal tersebut tiap relasinya hanya direpresentasikan secara generik, Artinya, relasi tetap akan memiliki karakteristik atau cara implementasi yang berbeda tergantung dari konsep awal dari arsitektur dibentuk. Berikut merupakan tabel relasi dari Archimate

Tabel 3.1Relasi Archimate

|  |  |  |  |
| --- | --- | --- | --- |
| ***Structural Relationships*** |  | ***Notation*** | ***Role Names*** |
| *Composition* | Mewakili bahwa suatu elemen terdiri dari satu atau lebih konsep lain. |  | *→ composed of ← composed in* |
| *Aggregation* | Mewakili bahwa suatu elemen menggabungkan satu atau lebih konsep lain. |  | *→ aggregates ← aggregated in* |
| *Assignment* | Mewakili alokasi tanggung jawab, kinerja perilaku, penyimpanan, atau eksekusi. |  | *→ assigned to ← has assigned* |
| *Realization* | Mewakili bahwa suatu elemen memainkan peran penting dalam penciptaan, pencapaian, rezeki, atau pengoperasian elemen yang lebih abstrak. |  | *→ realizes ← realized by* |
| *Dependency Relationships* |  | *Notation* | *Role Names* |
| *Serving* | Menyatakan bahwa suatu elemen menyediakan fungsinya ke elemen lain. | *image* | *→ serves ← served by* |
| *Access* | Mewakili kemampuan perilaku dan elemen struktur aktif untuk mengamati atau bertindak atas elemen struktur pasif. | *image* | *→ accesses ← accessed by* |
| *Influence* | Mewakili bahwa suatu elemen mempengaruhi implementasi atau pencapaian beberapa elemen motivasi. | *fig26* | *→ influences ← influenced by* |
| *Association* | Mewakili hubungan yang tidak ditentukan, atau hubungan yang tidak diwakili oleh hubungan ArchiMate lain. | *image* | *associated with → associated to ← associated from* |
| *Dynamic Relationships* |  | *Notation* | *Role Names* |
| *Triggering* | Mewakili hubungan temporal atau kausal antar elemen. |  | *→ triggers ← triggered by* |
| *Flow* | Mewakili transfer dari satu elemen ke elemen lainnya. |  | *→ flows to ← flows from* |
| *Other Relationships* |  | *Notation* | *Role Names* |
| *Specialization* | Menyatakan bahwa elemen adalah jenis tertentu dari elemen lain. |  | *→ specializes ← specialized by* |
| *Relationship Connectors* |  | *Notation* | *Role Names* |
| *Junction* | Digunakan untuk menghubungkan hubungan dari jenis yang sama. | *image* |  |

## Analisis Notasi dan Relasi *Layer* Arsitetkur

Pada lapisan/*layer* arsitektur sistem infromasi terdapat berbagai 2 bentuk relasi secara umum, relasi internal yaitu antar *element* pada lapisan sistem informasi dan relasa eksternal antara *element* pada lapisan arsitektur sistem informasi dengan lapisan arsitektur lain (1 lapisan sebelum atau sesudahnya).

Tabel 3.2 Mapping relasi antar element pada lapisan arsitektur sistem informasi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | | Application | | | | | | | | |
|  |  |  |  |  |  |  |  |  |
| Application |  | Application Component | cfgorstv | fotv | cfgortv | fiortv | fiortv | fiortv | fiortv | fiortv | ao |
|  | Application Collaboration | fgortv | cfgostv | cfgortv | fiortv | fiortv | fiortv | fiortv | fiortv | ao |
|  | Application Interface | fotv | fotv | cfgostv | fotv | fotv | fotv | fotv | fiotv | ao |
|  | Application Function | fotv | fotv | fotv | cfgostv | cfgotv | cfgotv | fotv | fiotv | ao |
|  | Application Interaction | fotv | fotv | fotv | cfgotv | cfgostv | cfgotv | fotv | fiotv | ao |
|  | Application Process | fotv | fotv | fotv | cfgotv | cfgotv | cfgostv | fotv | fiotv | ao |
|  | Application Event | fotv | fotv | fotv | fotv | fotv | fotv | cfgostv | fotv | ao |
|  | Application Service | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgostv | ao |
|  | Data Object | o | o | o | o | o | o | o | o | cgos |

Tabel 3.3 Mapping relasi antar element pada lapisan arsitektur sistem informasi dan teknologi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | | Technology | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Application |  | Application Component | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Collaboration | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Interface | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Function | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Interaction | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Process | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Event | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Application Service | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | ao | fotv | fotv | fotv | ao |
|  | Data Object | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o | o |

Tabel 3.2 menggambarkan hubungan antara elemen-elemen dalam lapisan arsitektur sistem informasi sementara Tabel 3.3 memberikan gambaran tentang bagaimana elemen-elemen tersebut berinteraksi dalam lapisan arsitektur sistem informasi dan teknologi. Setiap elemen memiliki relasi yang spesifik dengan elemen lainnya. Pada sisi kiri tabel, kita dapat melihat lapisan serta elemen-elemen awal, sementara di bagian atasnya terdapat lapisan dan elemen tujuan. Oleh karena itu, perpotongan antara baris dan kolom mewakili relasi yang digunakan untuk pasangan elemen tersebut.

Pada lapisan/*layer* arsitektur sistem infromasi terdapat berbagai 2 bentuk relasi secara umum, relasi internal yaitu antar *element* pada lapisan sistem informasi dan relasa eksternal antara *element* pada lapisan arsitektur sistem informasi dengan lapisan arsitektur lain (1 lapisan sebelum atau sesudahnya).

Tabel 3.4 Mapping relasi antar element pada lapisan arsitektur teknologi

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | | Technology | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Technology |  | Node | cfgiostv | cfgiortv | cfgiortv | fotv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | cfgiortv | cfgiotv | fotv | aio |
|  | Device | fotv | cfgostv | cfgiortv | fotv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fotv | fotv | fotv | ao |
|  | System Software | fotv | fotv | cfgiorstv | fotv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fotv | fotv | fotv | ao |
|  | Technology Collaboration | fgiotv | fgiortv | fgiortv | cfgostv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fgiortv | fgiotv | fotv | aio |
|  | Technology Interface | fotv | fotv | fotv | fotv | cfgostv | fotv | fotv | fotv | fotv | fotv | fotv | fiotv | aio | fotv | fotv | fotv | ao |
|  | Path | fgiotv | fgiortv | fgiortv | fgotv | fgiortv | cfgostv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fgiortv | fgiotv | fotv | aio |
|  | Communication Network | fortv | fgortv | fgiortv | fortv | fgiortv | fortv | cfgostv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fortv | fortv | fotv | ao |
|  | Technology Function | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgostv | cfgotv | cfgotv | fotv | fortv | ao | fotv | fotv | fotv | ao |
|  | Technology Process | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgotv | cfgostv | cfgotv | fotv | fortv | ao | fotv | fotv | fotv | ao |
|  | Technology Interaction | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgotv | cfgotv | cfgostv | fotv | fortv | ao | fotv | fotv | fotv | ao |
|  | Technology Event | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgostv | fotv | ao | fotv | fotv | fotv | ao |
|  | Technology Service | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fotv | cfgostv | ao | fotv | fotv | fotv | ao |
|  | Artifact | o | o | or | o | or | o | o | or | or | or | or | or | cgors | o | o | o | o |
|  | Equipment | fotv | cfgortv | cfgiortv | fotv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | cfgorstv | fotv | fotv | aio |
|  | Facility | cfgiotv | cfgiortv | cfgiortv | fotv | cfgiortv | fotv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | cfgiortv | cfgiostv | fotv | aio |
|  | Distribution Network | fgiortv | fgiortv | fgiortv | fortv | fgiortv | fortv | fotv | fiortv | fiortv | fiortv | fiortv | fiortv | aio | fgiortv | fgiortv | cfgostv | aio |
|  | Material | o | or | or | o | or | o | o | or | or | or | or | or | or | or | o | o | cgors |

Tabel 3.5 Mapping relasi antar element pada lapisan arsitektur teknologi dan sistem informasi

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | | | Application | | | | | | | | |
|  |  |  |  |  |  |  |  |  |
| Technology |  | Node | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Device | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | System Software | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Technology Collaboration | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Technology Interface | fotv | fotv | fortv | fotv | fotv | fotv | fotv | fortv | ao |
|  | Path | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Communication Network | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Technology Function | fotv | fotv | fotv | fortv | fortv | fortv | fotv | fortv | ao |
|  | Technology Process | fotv | fotv | fotv | fortv | fortv | fortv | fotv | fortv | ao |
|  | Technology Interaction | fotv | fotv | fotv | fortv | fortv | fortv | fotv | fortv | ao |
|  | Technology Event | fotv | fotv | fotv | fotv | fotv | fotv | fortv | fotv | ao |
|  | Technology Service | fotv | fotv | fotv | fotv | fotv | fotv | fotv | fortv | ao |
|  | Artifact | or | or | or | or | or | or | or | or | or |
|  | Equipment | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Facility | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Distribution Network | fortv | fortv | fortv | fortv | fortv | fortv | fortv | fortv | ao |
|  | Material | or | or | or | or | or | or | or | or | or |

Tabel 3.4 menggambarkan hubungan antara elemen-elemen dalam lapisan arsitektur sistem informasi sementara Tabel 3.5 memberikan gambaran tentang bagaimana elemen-elemen tersebut berinteraksi dalam lapisan arsitektur sistem informasi dan teknologi. Setiap elemen memiliki relasi yang spesifik dengan elemen lainnya. Pada sisi kiri tabel, kita dapat melihat lapisan serta elemen-elemen awal, sementara di bagian atasnya terdapat lapisan dan elemen tujuan. Oleh karena itu, perpotongan antara baris dan kolom mewakili relasi yang digunakan untuk pasangan elemen tersebut.

Berdasarkan tabel 3.3 dan 3.5, jika kita membandingkan satu sepasang element dari arsitektur aplikasi dengan teknologi ataupun sebaliknya terdapat sedikit perbedaan dalam relasinya. Seperti pada ***Application Component → Node*** memiliki relasi ***fotv*** sementara sebaliknya, jika ***Node Application → Compoenent*** memiliki relasi ***fortv***yang menandakan bahwa terdapat perbedaan sudut pandang dari lapisan arsitektur. Seperti yang kita tahu bahwa dalam perancangan arsitektur menggunakan Archimate mengacu pada alur yang telah ditentukan seperti pada Gambar 2.2 membentuk sebauh *cycle* yang berkelanjutkan, namun dengan fakta yang kita dapatkan ini. Kita tidak bisa serta merta merancang notasi dan relasi, tanpa memperhitungkan ketergantungan komponen satu sama lain antar lapisan.

Ketika melihat dari ***Node → Application Component***, terjadi serangkaian interaksi yang mengindikasikan bagaimana infrastruktur teknologi berinteraksi dengan komponen aplikasi. E*lement* Node berfungsi sebagai tempat di mana komponen aplikasi diimplementasikan dan dijalankan baik itu server fisik, instance cloud, atau perangkat lainnya. Node menyediakan lingkungan yang diperlukan untuk menjalankan aplikasi, menyediakan sumber daya komputasi, penyimpanan, dan layanan lainnya yang diperlukan oleh komponen aplikasi. Selain itu, node juga dapat memicu atau memulai proses atau fungsi di dalam aplikasi, misalnya, dengan memproses permintaan atau menyediakan layanan. Hubungan asosiatif antara node dan komponen aplikasi menandakan bahwa node mendukung atau berhubungan dengan fungsi yang dijalankan oleh komponen aplikasi. Sebagai contoh, server web yang menjalankan aplikasi e-commerce berhubungan dengan komponen aplikasi yang mengelola proses transaksi dan menyediakan layanan kepada pengguna.

Sebaliknya, ketika melihat dari sudut pandang ***Application Component → Node***, interaksi antara Application Component dan infrastruktur node terjadi dalam konteks penggunaan dan pemanfaatan sumber daya infrastruktur. Application Component, seperti layanan web atau aplikasi bisnis, menghasilkan atau mengalirkan data dan informasi ke infrastruktur node. Node, sebagai penyedia lingkungan pelaksanaan, menerima data atau instruksi dari aplikasi dan memprosesnya sesuai kebutuhan. Application Component tergantung pada node untuk menyediakan sumber daya yang diperlukan untuk beroperasi, termasuk sumber daya komputasi, jaringan, dan penyimpanan. Node berfungsi sebagai layanan yang diperlukan oleh aplikasi untuk berjalan dengan lancar, dan aplikasi mungkin memicu atau memulai tindakan atau proses di dalam node sesuai dengan kebutuhan aplikasi. Hubungan antara Application Component dan node mencerminkan integrasi antara fungsionalitas aplikasi dan infrastruktur yang mendukungnya dalam lingkungan teknologi yang lebih luas.

## Analisis *Graph Database Neo4J*

## Implementasi

Pada tahap implementasi ini akan terbagi menjadi 3 bagian secara umum yaitu *Mapping* relasi elemen antar layer pada studi kasus, mendefinisikan *rules* dan pengujian menggunakan Neo4J, untuk detail dari tahapan yang dilakukan adalah sebagai berikut:

## *Mapping* Relasi ElemenAntar *Layer* Pada Studi Kasus

Dilakukan *mapping* pada dataset yang didapat dari kelas TKTI 2023/2024 dan studi kasus ArchIsurance dari TOGAF. Pada dataset yang didapat dilakukan ekstraksi kedalam bentuk csv, pada proses ekstraksi ini akan didapat 3 file yaitu *elements*, *relations* dan *properties*. File hasil ekstraksi yang akan digunakan hanya dua yaitu *elements* yang didalamnya terdapat atribut id, nama *element* dan nama label seperti pada . Sementara untuk file *relations* didalamnya terdapat atribut id, source\_id, relasi, dan target\_id. Pada file *relation* akan dilakukan pencocokan id dengan source\_id dan target\_id dengan file *element* untuk mendapatkan notasi element yang berpasangan, setelah itu baru akan dilukan pengecekan relasiberdasarkan Tabel 3.3 dan Tabel 3.5 untuk melihat apakah pasangan *element* memiliki relasi yang sesuai berdasarkan panduan yang ada. Hasil dari mapping relasi yang dilakukan adalah sebagai berikut :

1. ArchIsurance

## Mendefinisikan *Rules*.

## Pengujian menggunakan NEO4J

Dalam ArchiMate, lapisan arsitektur aplikasi dan teknologi saling berinteraksi dan berhubungan satu sama lain. Berikut adalah beberapa aturan untuk memetakan lapisan arsitektur aplikasi ke teknologi dalam ArchiMate:

[**Serving Relationships**: Hubungan “serving” dapat digunakan antara layanan teknologi dan berbagai jenis elemen perilaku aplikasi, dan antara antarmuka teknologi dan komponen aplikasi1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html).

[**Realization Relationships**: Hubungan “realization” dapat digunakan dari proses atau fungsi teknologi ke proses atau fungsi aplikasi, dari objek teknologi ke objek data, untuk menunjukkan bahwa objek data direalisasikan oleh, misalnya, file data fisik2](https://pubs.opengroup.org/architecture/archimate31-doc/chap12.html).

[**Application Component**: Komponen aplikasi mewakili enkapsulasi fungsionalitas aplikasi yang sejalan dengan struktur implementasi, yang modular dan dapat diganti3](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Application-Layer.html). Komponen aplikasi ini dapat direalisasikan oleh komponen teknologi.

[**Application Interface**: Dalam pengertian yang lebih luas, antarmuka aplikasi mendefinisikan beberapa karakteristik perilaku dasar: mendefinisikan set operasi dan peristiwa yang disediakan oleh komponen3](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Application-Layer.html). Antarmuka aplikasi ini dapat direalisasikan oleh antarmuka teknologi.

Harap dicatat bahwa aturan ini adalah panduan umum dan mungkin perlu disesuaikan tergantung pada kebutuhan spesifik Anda. [Selalu baik untuk merujuk ke dokumentasi ArchiMate resmi atau panduan lainnya untuk pemahaman yang lebih mendalam](https://www.visual-paradigm.com/guide/archimate/full-archimate-viewpoints-guide/" \t "_blank)[4](https://www.visual-paradigm.com/guide/archimate/full-archimate-viewpoints-guide/)[3](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Application-Layer.html)[2](https://pubs.opengroup.org/architecture/archimate31-doc/chap12.html)[1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html)[5](https://bizzdesign.com/blog/practical-archimate-viewpoints-for-the-application-layer/). Semoga ini membantu! Jika Anda memiliki pertanyaan lain, jangan ragu untuk bertanya. 😊

|  |  |  |
| --- | --- | --- |
| No | Materi | Link |
| 1 |  |  |
| 2 | Relationship and Connector | [Relationships and Relationship Connectors: ArchiMate® 3.2 Specification (opengroup.org)](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-and-Relationship-Connectors.html) |
|  | Alignment of the Application and Technology Layers | [Relationshops between Core Layers: ArchiMate® 3.2 Specification (opengroup.org)](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html) |
|  |  | [Full ArchiMate Viewpoints Guide (Examples Included) (visual-paradigm.com)](https://www.visual-paradigm.com/guide/archimate/full-archimate-viewpoints-guide/#layered-viewpoint) |
|  |  | [Language Notation Guide (Clickable): ArchiMate® 3.2 Specification (opengroup.org)](https://pubs.opengroup.org/architecture/archimate3-doc/language-notation-guide.html) |

Berikut adalah beberapa aturan notasi ArchiMate untuk memetakan lapisan arsitektur aplikasi ke teknologi dan sebaliknya:

**Dari Aplikasi ke Teknologi**:

[**Serving Relationships**: Hubungan “serving” dapat digunakan antara layanan teknologi dan berbagai jenis elemen perilaku aplikasi, dan antara antarmuka teknologi dan komponen aplikasi1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html).

[**Realization Relationships**: Hubungan “realization” dapat digunakan dari proses atau fungsi aplikasi ke proses atau fungsi teknologi1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html).

**Dari Teknologi ke Aplikasi**:

[**Serving Relationships**: Hubungan “serving” dapat digunakan antara layanan aplikasi dan berbagai jenis elemen perilaku teknologi, dan antara antarmuka aplikasi dan elemen struktur aktif internal teknologi1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html).

[**Realization Relationships**: Hubungan “realization” dapat digunakan dari proses atau fungsi teknologi ke proses atau fungsi aplikasi1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html).

Harap dicatat bahwa aturan ini adalah panduan umum dan mungkin perlu disesuaikan tergantung pada kebutuhan spesifik Anda. [Selalu baik untuk merujuk ke dokumentasi ArchiMate resmi atau panduan lainnya untuk pemahaman yang lebih mendalam](https://pubs.opengroup.org/architecture/archimate3-doc/language-notation-guide.html" \t "_blank)[2](https://pubs.opengroup.org/architecture/archimate3-doc/language-notation-guide.html)[1](https://pubs.opengroup.org/architecture/archimate3-doc/ch-Relationships-Between-Core-Layers.html)[3](https://pubs.opengroup.org/architecture/archimate31-doc/chap12.html)[4](https://digital-portfolio.opengroup.org/archimate-specs/latest/11-relationships-between-core-layers/relationships-between-core-layers.html). Semoga ini membantu! Jika Anda memiliki pertanyaan lain, jangan ragu untuk bertanya. 😊

# BAB IV HASIL DAN PEMBAHASAN

## Hasil Pengujian

## Pembahasan/Diskusi

(analisis, sintesis, dan evaluasi)

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Berupa hasil penelitian/perancangan yang menjawab permasalahan atau yang berupa konsep, program, dan karya rancangan

## Saran (jika dianggap perlu)

berisi hal-hal yang masih dapat dikerjakan dengan lebih baik dan dapat dikembangkan lebih lanjut, atau berisi masalahmasalah yang dialami pada saat proses pengerjaan tugas/proyek akhir.

# DAFTAR PUSTAKA

[1] D. Naranjo, M. Sànchez, and J. Villalobos, “PRIMROSe: A graph-based approach for enterprise architecture analysis,” *Lect. Notes Bus. Inf. Process.*, vol. 227, no. May 2019, pp. 434–452, 2015, doi: 10.1007/978-3-319-22348-3\_24.

[2] M. Smajevic and B. Dominik, “Towards Graph-based Analysis of Enterprise Architecture Models,” no. October, pp. 199–209, 2021.

[3] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice Second Edition Third Edition*. 2013. [Online]. Available: https://www.oreilly.com/library/view/software-architecture-in/9780132942799/

[4] M. Hakim, M. Ghozali, W. Sugiharto, R. Nindyasari, and A. Jazuli, “Information Systems Identification Using TOGAF Framework: A Study Case at PT. X,” pp. 117–124, 2019, doi: 10.4108/eai.24-10-2018.2280638.

[5] H. Rachel, “TOGAF® 9 Foundation Study Guide 3rd Edition.” [Online]. Available: www.vanharen.net

[6] A. Buchalcevova, “Using ArchiMate to model ISO/IEC 29110 standard for very small entities,” *Comput. Stand. Interfaces*, vol. 65, no. February, pp. 103–121, 2019, doi: 10.1016/j.csi.2019.03.001.

[7] The Open Group, “ArchiMate® 3.1 Specification,” The Open Group. [Online]. Available: https://pubs.opengroup.org/architecture/archimate31-doc/

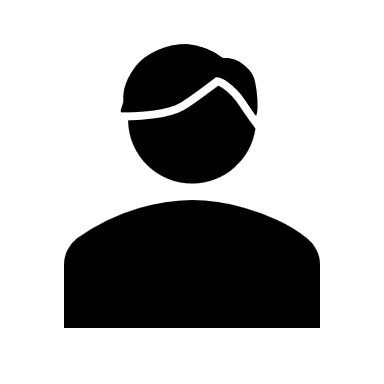
[8] The Open Group, *ArchiSurance Case Study*. 2012, p. 32. [Online]. Available: https://vdocument.in/download/archisurance-case-study.html

[9] M. Saad, Y. Zhang, J. Tian, and J. Jia, “A graph database for life cycle inventory using Neo4j,” *J. Clean. Prod.*, vol. 393, no. January, p. 136344, 2023, doi: 10.1016/j.jclepro.2023.136344.

[10] Neo4J, “Neo4J Sandbox.” [Online]. Available: https://neo4j.com/sandbox/

# LAMPIRAN-LAMPIRAN ATAU APPENDIKS (jika ada)

# BIODATA PENULIS

Penulis dilahirkan di Madiun, 29 Januari 1985, merupakan anak pertama dari 4 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK ABA 18 Madiun, SDN Beteng 1 Madiun, SMPN 2 Madiun dan SMAN 2 Madiun. Setelah lulus dari SMAN tahun 2020, Penulis mengikuti SBMPTN dan diterima di Departemen Teknik Mesin FTIRS - ITS pada tahun 2020 dan terdaftar dengan NRP 02112040000130.

Di Departemen Teknik Mesin Penulis sempat aktif di beberapa kegiatan Seminar yang diselenggarakan oleh Departemen, Himpunan Mahasiswa Teknik Mesin (HMM) dan aktif sebagai Asisten Praktikum Mesin Konversi Enersi maupun Grader mata kuliah Termodinamika.