**SW Technical Document**

**Implementasi *Kafka***

**Disusun Oleh:**

| 11S21007 | Tiara Yuliandari |
| --- | --- |
|  |  |

**Untuk:**

**PT. Federal International Finance**

**Jakarta**

|  | | **Kerja Praktek 2014**  **Institut Teknologi Del** | | | |  |
| --- | --- | --- | --- | --- | --- | --- |
| *No. Dokumen: SW-KP-24-120-D* | | | *Versi: xx.xx* | *Tanggal : DD-MM-YY* | *Jumlah Halaman :* | |

**Lembar Pengesahan**

**Dokumen Teknis**

**Proyek Sistem Informasi**

Implementasi *Kafka*

**Oleh:**

| NIM 11S21007 | Tiara Yuliandari |
| --- | --- |
|  |  |

Sitoluama, 6 Juni 2012

| Pembimbing |
| --- |
| Dr. Johannes Harungguan Sianipar, S.T., M.T.  NIDN. 0116047301 |

**DAFTAR ISI**

[**1 Introduction 5**](#)

[**1.1 Purpose of Document 5**](#)

[**1.2 Scope 5**](#)

[**1.3 Identification and Numbering 6**](#)

[**1.4 Reference Documents 6**](#)

[**1.5 Document Summary 6**](#)

[**2 System Overview 8**](#)

[**2.1 Current System Overview 8**](#)

[**3 Software General Description 10**](#)

[**3.1 SW Environment 10**](#)

[3.1.1 Development Environment 10](#_3rdcrjn)

[**3.1.2 Operational Environment 11**](#)

[**4 Requirement Definition 12**](#_lnxbz9)

[4.1 External Interface 12](#_35nkun2)

[**4.1.1 Hardware Interface 12**](#)

[4.2 Functional Description 12](#_393o0n26xse1)

[4.3 Non Functional Requirement 12](#_1ksv4uv)

[**4.4 Design Constraint 13**](#)

[**4.5 Sequence Diagram 14**](#)

[**4.5.1 Use Case: sync data stream 14**](#)

[**5 SW Item Description & Installation 16**](#)

[**5.1.1 SW Item & Location 16**](#)

[**5.1.2 SW Installation 16**](#)

[**5.1.3 Precondition 20**](#)

[**5.1.4 Procedures (Working Instruction) 20**](#)

[**5.1.5 Reporting 20**](#)

[**LAMPIRAN 21**](#)

[**Sejarah Versi 22**](#)

[**Sejarah Perubahan 23**](#)

# **Introduction**

Dokumen ini merupakan dokumen teknis yang dibuat dengan tujuan untuk mendokumentasikan pengerjaan proyek. Dokumen ini meliputi pendefinisian kebutuhan, analisis, design, implementasi, hingga testing. Dengan adanya dokumen ini, diharapkan dapat memperjelas dan menyatukan pemahaman tentang proyek.

## **Purpose of Document**

Dokumen ini ditujukan sebagai panduan teknis dari proyek implementasi *kafka* yang ditugaskan pada mahasiswa pelaksana. Dokumen ini akan memuat penjelasan proyek, pendekatan yang akan digunakan, hingga hasilnya yang akan dijelaskan secara umum. Dokumen ini ditulis oleh untuk para pengembang aplikasi, dan ditulis berdasarkan spesifikasi kebutuhan “pemilik” aplikasi yaitu unit organisasi yang akan mengoperasikan aplikasi yang dibuat.

## **Scope**

Dokumen ini akan mencakup deskripsi umum proyek, pendekatan yang digunakan dalam pengimplementasian proyek, hingga deskripsi kebutuhan selama pengerjaan proyek. Dokumen ini tidak akan secara langsung memberikan deskripsi asli dari proyek yang akan dikerjakan oleh mahasiswa. Hal ini perlu dilakukan dikarenakan perlunya untuk menjaga keamanan dan privasi dari Perusahaan.

Definition, Akronim and *Abbreviation*

Berikut merupakan daftar definisi, singkatan dan penjelasan dari setiap istilah yang digunakan pada dokumen teknis ini.

| **No** | **Definisi** | **Penjelasan** |
| --- | --- | --- |
|  | ToR | *Term of Reference* |
|  | MoM | *Minutes of Meeting* |
|  | PIP | *Project Implementation Plan* |
|  | SW | Software |
|  | LA | *Log Activity* |
|  | KP | Kerja Praktek |

## **Identification and Numbering**

Semua identifikasi dan tatacara penomoran mengacu ke Std-PA2-09

## **Reference Documents**

Berikut merupakan dokumen yang digunakan sebagai rujukan pembuatan dokumen ini.

1. ToR-KP-24-120-D  
   berisikan kerangka awal kerja yang akan dilaksanakan. Dokumen ini mencakup deskripsi proyek dan periode pengerjaan yang akan dilaksanakan oleh mahasiswa pelaksana.
2. PIP-KP-24-120-D  
   Berisikan kerangka awal pekerjaan yang akan dilaksanakan oleh mahasiswa pelaksana. Dokumen ini meliputi deskripsi umum proyek, hingga manajemen proyek dan *timeline* pengerjaan.

## **Document Summary**

Dokumen ini terdiri dari lima bab yang berisikan penjelasan dan kebutuhan-kebutuhan yang harus diperhatikan selama proses pelaksanaan kerja praktek. berikut adalah penjelasan isi dari setiap bab pada dokumen ini.

**Bab 1 Introduction**, berisikan tujuan dan latar belakang ditulisnya dokumen ini. Selain dari itu, pada bab ini dijelaskan pula ruang lingkup dari dokumen ini, penjelasan definisi, akronim dan singkatan, kemudian penomoran dalam dokumen, dokumen yang menjadi referensi, serta ringkasan isi dokumen.

**Bab 2 System Overview**, berisikan deskripsi mengenai sistem yang sekarang dikembangkan pada perusahaan.

**Bab 3 Software General Description**, berisikan penjelasan sistem yang akan dikerjakan serta lingkungan pengembangan.

**Bab 4 Requirement Definition,** berisikan penjelasan kebutuhan sehingga sistem dikatakan berhasil. Kebutuhan ini meliputi kebutuhan antarmuka, kebutuhan fungsional, kebutuhan non fungsional, keterbatasan design, hingga urutan kerja sistem yang dikembangkan.

**Bab 5 SW Item Description & Installation,** berisikan penjelasan mengenai perangkat lunak yang digunakan.

# ***System Overview***

Pengimplementasian *kafka* pada proyek yang ada bertujuan untuk meningkatkan performansi dari proses *consuming* dan *inserting* pada link *existing project.* Hal ini diperlukan dikarenakan Ketika proses *consuming* dan *inserting* membutuhkan waktu yang cukup lama Ketika data yang di-*consume* dari topik cukup banyak.

*Kafka* adalah *middleware* yang saat ini digunakan oleh FIFGROUP untuk pertukaran pesan berukuran besar. *Kafka* dipilih agar pertukaran pesan menjadi lebih cepat dan andal. Dalam ekosistem *Kafka*, terdapat *publisher* yang menghasilkan data, *Kafka* menyimpan dan mendistribusikannya, sementara *consumer* memproses data tersebut. Hal ini memungkinkan komunikasi dan aliran data yang efisien antar berbagai sistem.

Fitur unggulan *Kafka* adalah *Kafka Streams*, sebuah *library client-side* yang memungkinkan pembangunan aplikasi *streaming* yang tangguh dan dapat diskalakan. *Kafka Streams* dapat melakukan transformasi dan analisis data secara *real-time*, serta mendukung *stateful processing* untuk agregasi dan perhitungan kompleks, sehingga aplikasi dapat bereaksi terhadap perubahan data secara instan. Sifat *real-time* ini juga membuat *Kafka Streams* memiliki performa yang lebih baik saat memproses data berukuran besar.

Saat ini, FIFGROUP hanya memanfaatkan *publisher* dan *consumer* yang disediakan oleh *Kafka* untuk memproses hampir semua komunikasi pesan, termasuk pesan berukuran sangat besar, salah satunya pada *current system*. Hal ini menyebabkan masalah krusial, yaitu pemrosesan data yang lambat dan memakan waktu berjam-jam. Oleh karena itu, proyek ini bertujuan untuk meningkatkan performa pemrosesan pesan menggunakan *Kafka Streams* berdasarkan keunggulan-keunggulan yang dimilikinya.

## **Current System Overview**

*Current system* yang digunakan saat ini merupakan proyek SpringBoot yang sudah memanfaatkan *kafka* untuk bertukar data antara Perusahaan dan Perusahaan pihak ketiga. Pada mulanya *FIFGROUP* akan mengirim pesan ke topik menggunakan *kafka publisher* lalu Perusahaan pihak ketiga akan mengirimkan balasan. Balasan yang diberikan akan berada di topik yang nantinya akan di-*consume* menggunakan *kafka consumer.* Penggunaan *kafka* pada *current system* masih cukup umum, yaitu hanya menggunakan satu *publisher* dan satu *consumer* dalam pertukaran *messages-*nya. Pada proses pertukaran data pula, *messages* yang dikirimkan cukup besar, sehingga hal ini akan membutuhkan waktu yang cukup lama untuk proses *consuming* serta *inserting* messagenya.

# **Software General Description**

Untuk penanganan masalah yang terjadi pada *current system* diperlukan peningkatan dalam implementasi *kafka*. Pendekatan yang digunakan untuk meningkatkan performansi sistem yang digunakan saat ini adalah dengan menggunakan beberapa *consumer* dan *publisher (staging kafka)* serta mengimplementasikan *Kafka* *Stream* untuk proses *streaming* datanya. Proses ini dimulai dari *splitting* data, *filter* dan *mapping* data pada *kafka* *streams* *topology*, hingga proses *inserting* data ke tabel. Pada topologi yang dibangun menggunakan *Kafka Streams* terdapat proses sinkronisasi dengan melakukan compose data sesuai dengan tipe yang digunakan pada database FIFGROUP.

## **SW Environment**

### **Development Environment**

Berikut merupakan lingkungan pengembangan yang meliputi spesifikasi *hardware* dan *software.*

| *No* | *Hardware* | *Spesifikasi* |
| --- | --- | --- |
| 1 | Processor | Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.59 GHz |
| 2 | RAM | 16.00 GB |

| *No* | *Software* | *Spesifikasi* |
| --- | --- | --- |
| 1 | Sistem Operasi | Windows |
| 2 | Paket Office | Microsoft Office 2016 |
| 3 | Tools Editor | IntelliJ IDEA |
| 4 | Database | Oracle SQL |
| 5 | Web Server | nginX |
| 6 | Framework | SpringBoot |
| 7 | Middleware | Apache Kafka |
| 8 | Testing Tools | Postman, JMeter |
| 9 | Lainnya | Docker |

### **Operational Environment**

Berikut merupakan lingkungan operasional yang meliputi spesifikasi *hardware* dan *software.*

| *No* | *Hardware* | *Spesifikasi* |
| --- | --- | --- |
| 1 | Processor | Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.59 GHz |
| 2 | RAM | 16.00 GB |

| *No* | *Software* | *Spesifikasi* |
| --- | --- | --- |
| 1 | Sistem Operasi | Windows |
| 2 | Paket Office | Microsoft Office 2016 |
| 3 | Tools Editor | IntelliJ IDEA |
| 4 | Database | Oracle SQL |
| 5 | Web Server | nginX |
| 6 | Framework | SpringBoot |
| 7 | Middleware | Apache Kafka |
| 8 | Testing Tools | Postman, JMeter |
| 9 | Lainnya | Docker |

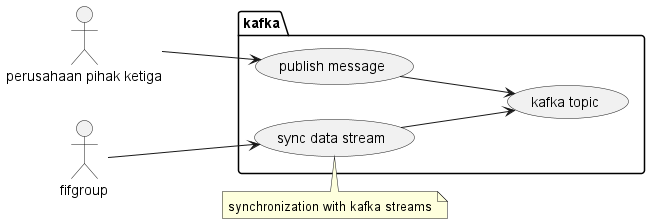
# **Requirement Definition**

## **External Interface**

### **Hardware Interface**

| *No* | *Hardware* | *Spesifikasi* |
| --- | --- | --- |
| 1 | Processor | Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.59 GHz |
| 2 | RAM | 16.00 GB |

## Functional Description

**

Gambar di atas merupakan use case diagram yang menggambarkan proses dari proyek ini. Ketika terdapat message pada topik maka fifgroup akan melakukan *data synchronization* dengan memanfaatkan *kafka streams.*

## **Non Functional Requirement**

| **SRS-Id** | **Parameter** | **Requirement** |
| --- | --- | --- |
| SRS-01 | Availability | sistem harus mampu beroperasi terus menerus tanpa gagal. |
| SRS-02 | Reliability | sistem harus mampu melakukan sinkronisasi data tanpa ada kegagalan. Artinya seluruh data harus di-*mapping* dengan benar dan berhasil. |
| SRS-03 | Response time | sistem harus mampu melakukan data sinkronisasi lebih cepat dari *current system.* |

## **Design Constraint**

Pada pengerjaan proyek ini terdapat beberapa hal yang harus diperhatikan yaitu sebagai berikut.

1. Telah menginstall Maven.
2. Telah menginstal java versi 17.
3. mengetahui MVC dan organisasi proyek pada Spring Boot.

## **Sequence Diagram**

### Use Case: sync data stream

#### 

Proses data sinkronisasi ini merupakan proses *mapping* dan *inserting data* yang dimulai dari *splitting data* berdasarkan tipe datanya yaitu *single* atau *array data type* kemudian akan dilakukan pengecekan berdasarkan *aggregator type,* yaitu *company* dan *individual.* selanjutnya sistem akan mendapatkan *list* tabel yang sesuai dengan *aggregator type* dan kemudian akan di-*compose* di dalam *topology*. Data yang telah di-*compose* akan dikirimkan kembali ke topik *pfx-stream* untuk *labelling* sesuai dengan *aggregator type* lalu dikirim ke *pfx-process-notif* untuk *labelling* sesuai dengan tipe datanya.*.* Lalu dikirimkan kembali ke topik yang selanjutnya akan dilanjutkan proses *insertion* dan *updating flag.*

#### **Algorithm**

**01 : sync data stream**

**Objek terkait : sinkronisasi Data**

**Event :**

**Initial State (IS)**: data belum sinkronisasi

**Final State (FS)**: data berhasil di sinkronisasi

**Spesifikasi Proses/algoritma**:

1. Cek *aggregator type*
2. IF nama tabel yang sesuai dengan *aggregator type  
   THEN compose data stream* di *topology*ELSE proses dihentikan
3. *Publish* ke kafka topik
4. *compose label*
5. IF DataType adalah SINGLE/ARRAY  
   THEN process data  
   ELSE proses dihentikan
6. *update flag process*

Catatan : jika satu layar mengandung banyak button, dan untuk satu button ditulis sebuah prosedur, maka akan ada beberapa algoritma

#### **Spesifikasi Report**

Spesifikasi report akan dilampirkan.

# **SW Item Description & Installation**

### **SW Item & Location**

SW item yang akan diberikan merupakan prototipe kode program beserta dokumentasinya dalam bentuk zip. File akan dinamai deliverables-KP-24-120-D.zip. Item yang dilampirkan pada file tersebut adalah sebagai berikut.

1. Dokumen Administrasi:  
   TOR : deliverables\dokumen-administrasi\ToR-KP-24-120-D.pdf  
   PIP : deliverables\dokumen-administrasi\PIP-KP-24-120-D.pdf
2. Dokumen Teknis:  
   SW : deliverables\dokumen-teknis\SW-KP-24-120-D.pdf
3. Laporan Kerja Praktek:  
   REP : deliverables\laporan-kerja-praktik\REP-KP-24-120-D.pdf
4. Source Code: deliverables\source-code\pfxkafkaprototype.zip

### **SW Installation**

Hal yang perlu dilakukan untuk instalasi proyek ini adalah dengan menginstal openjdk dan maven. berikut merupakan langkah-langkahnya.

#### 5.1.2.1 Langkah-langkah Instalasi JDK 17

##### **Mengunduh JDK 17**

1. Buka browser dan kunjungi situs resmi penyedia JDK, yaitu [Oracle JDK](https://www.oracle.com/java/technologies/javase-jdk17-downloads.html) atau [AdoptOpenJDK](https://adoptopenjdk.net/).
2. Pilih versi JDK 17 yang sesuai dengan sistem operasi Windows (x64).
3. Klik tombol unduh untuk mendapatkan file installer dalam format .msi atau .zip.

##### **Menginstal JDK 17**

1. **Menggunakan Installer .msi:**
   * Jalankan file installer .msi yang telah diunduh.
   * Ikuti petunjuk pada layar. Pada tahap ini, pengguna dapat memilih lokasi instalasi atau menggunakan lokasi default (C:\Program Files\Java\jdk-17).
   * Setelah selesai, klik tombol "Finish" untuk menyelesaikan proses instalasi.
2. **Menggunakan File .zip:**
   * Ekstrak file .zip ke direktori yang diinginkan, misalnya C:\Program Files\Java\jdk-17.
   * Tidak diperlukan langkah tambahan karena file sudah siap digunakan setelah diekstrak.

##### **Menetapkan Variabel Lingkungan JAVA\_HOME**

1. Buka **Control Panel** > **System** > **Advanced System Settings**.
2. Pada jendela **System Properties**, pilih tab **Advanced** dan klik tombol **Environment Variables**.
3. Di bagian **System Variables**, klik tombol **New** dan tambahkan:
   * **Variable name:** JAVA\_HOME
   * **Variable value:** Jalur instalasi JDK, misalnya C:\Program Files\Java\jdk-17.
4. Klik **OK** untuk menyimpan variabel baru ini.

##### **Menambahkan JDK ke Path**

1. Dalam jendela **Environment Variables**, temukan dan pilih variabel **Path** di bawah **System Variables**. Klik tombol **Edit**.
2. Klik tombol **New** dan tambahkan jalur bin dari instalasi JDK, misalnya:
   * C:\Program Files\Java\jdk-17\bin
3. Klik **OK** untuk menyimpan perubahan.

##### **Verifikasi Instalasi**

1. Buka Command Prompt dengan menekan Windows + R, ketik cmd, dan tekan Enter. Ketik perintah berikut untuk memverifikasi instalasi: java -version
2. Jika instalasi berhasil, akan muncul output seperti berikut:  
   java version "17.0.x"

Java(TM) SE Runtime Environment (build 17.0.x)

Java HotSpot(TM) 64-Bit Server VM (build 17.0.x, mixed mode, sharing)

**5.1.2.2 Langkah-langkah Instalasi Maven**

##### **Mengunduh Apache Maven**

1. Buka browser dan kunjungi situs resmi [Apache Maven](https://maven.apache.org/download.cgi).
2. Pada halaman unduhan, pilih versi terbaru yang stabil (biasanya dalam format tar.gz atau zip) dan klik tautan unduh untuk mendapatkan file.
3. Setelah unduhan selesai, simpan file di direktori yang diinginkan.

##### **Ekstrak File**

1. Jika file yang diunduh dalam format .zip, ekstrak file tersebut ke direktori pilihan, misalnya C:\Program Files\Apache\maven.

Struktur direktori hasil ekstraksi seharusnya terlihat seperti ini:  
C:\Program Files\Apache\maven\apache-maven-3.8.x

##### **Menetapkan Variabel Lingkungan MAVEN\_HOME**

1. Buka **Control Panel** > **System** > **Advanced System Settings**.
2. Pada jendela **System Properties**, pilih tab **Advanced** dan klik tombol **Environment Variables**.
3. Di bagian **System Variables**, klik tombol **New** dan tambahkan:
   * **Variable name:** MAVEN\_HOME
   * **Variable value:** Jalur direktori hasil ekstraksi Maven, misalnya C:\Program Files\Apache\maven\apache-maven-3.8.x.
4. Klik **OK** untuk menyimpan variabel baru ini.

##### **Menambahkan Maven ke Path**

1. Dalam jendela **Environment Variables**, temukan dan pilih variabel **Path** di bawah **System Variables**. Klik tombol **Edit**.
2. Klik tombol **New** dan tambahkan jalur bin dari direktori Maven, misalnya:
   * C:\Program Files\Apache\maven\apache-maven-3.8.x\bin
3. Klik **OK** untuk menyimpan perubahan.

##### **Verifikasi Instalasi**

1. Buka Command Prompt dengan menekan Windows + R, ketik cmd, dan tekan Enter. Ketik perintah berikut untuk memverifikasi instalasi Maven: mvn -v
2. Jika instalasi berhasil, akan muncul output yang menampilkan informasi versi Maven, JDK, dan sistem operasi, seperti: Apache Maven 3.8.x (XXXXXXXXXXXXX)  
   Maven home: C:\Program Files\Apache\maven\apache-maven-3.8.x  
   Java version: 17.0.x, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-17  
   Default locale: en\_US, platform encoding: Cp1252  
   OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

### **Precondition**

precondition yang harus dipenuhi adalah laptop sudah dilengkapi kafka sehingga dapat menjalankan proyeknya.

### **Procedures (Working Instruction)**

Prosedur menjalankan proyek ini adalah dengan langkah langkah sebagai berikut.  
1. Jalankan perintah: mvn clean package

2. Selanjutnya, Jalankan jar yang ada pada folder target.

### **Reporting**

Proses reporting selama kerja praktek ini dilakukan setiap hari untuk pengecekan progress-nya. pelaporan proyek dilakukan bersama mentor atau pembimbing yang berada di perusahaan yaitu Bapak Raden Raenovaldy Akhmad E. Pelaporan final proyek dilakukan selambat-lambatnya seminggu sebelum Kerja Praktek selesai.

# **LAMPIRAN**

Deliverables :

<https://github.com/TiaraYu8/kafkaImplementation.git>

# **Sejarah Versi**

*Pada bagian ini, dijelaskan semua versi yang pernah di-deliver, dan ciri serta perubahannya. Untuk Kerja praktek ini, minimal ada dua versi : versi pada saat presentasi I, dan versi final.*

| **Versi** | **Ditulis Oleh** | **Tanggal** | **Disetujui Oleh** | **Tanggal** |
| --- | --- | --- | --- | --- |
| Draft |  |  | Pembimbing |  |
| Final |  |  | Pembimbing |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# **Sejarah Perubahan**

*Bagian ini memuat sejarah perubahan dokumen (no. versi terbaru dibandingkan versi sebelumnya).*

**No. dokumen :**

**No. versi :**

| **Halaman** | **Semula** | **Menjadi** | **Alasan perubahan** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**No. dokumen :**

**No. versi :**

| **Halaman** | **Semula** | **Menjadi** | **Alasan perubahan** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |