



KERJA PRAKTIK - IF184801

**IMPLEMENTASI DOCKER UNTUK PENGELOLAAN
APLIKASI SIAKAD PT SENTRA VIDYA UTAMA (SEVIMA)**

PT Sentra Vidya Utama

Medokan Asri Tengah MA-2 Blok Q No 16, Medokan Ayu, Kec.
Rungkut, Kota SBY, Jawa Timur 60295

Periode: 1 September 2021 - 28 Februari 2022

Oleh:

Restu Agung Parama

05111840000123

Pembimbing Jurusan

Ir. F.X. ARUNANTO, M.Sc.

Pembimbing Lapangan

Anandre Forastero, S.Psi., M.Psi., Psikolog

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2022



KERJA PRAKTIK - IF184801

**IMPLEMENTASI DOCKER UNTUK PENGELOLAAN
APLIKASI SIAKAD PT SENTRA VIDYA UTAMA (SEVIMA)**

PT Sentra Vidya Utama

Medokan Asri Tengah MA-2 Blok Q No 16, Medokan Ayu, Kec.
Rungkut, Kota SBY, Jawa Timur 60295

Periode: 1 September 2021 - 28 Februari 2022

Oleh:

Restu Agung Parama

05111840000123

Pembimbing Jurusan

Ir. F.X. Arunanto, M.Sc.

Pembimbing Lapangan

Anandre Forastero, S.Psi., M.Psi., Psikolog

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2022

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iv
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
LEMBAR PENGESAHAN	xi
KATA PENGANTAR	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat	2
1.4 Rumusan Masalah	2
1.5 Lokasi dan Waktu Kerja Praktik	2
1.6 Metodologi Kerja Praktik	3
1.6.1 Perumusan Masalah	3
1.6.2 Studi Literatur	3
1.6.3 Perancangan Sistem	3
1.6.4 Implementasi Sistem	3
1.6.5 Pengujian dan Evaluasi	3
1.6.6 Kesimpulan dan Saran	4
1.7 Sistematika Laporan	4
1.7.1 Bab I Pendahuluan	4
1.7.2 Bab II Profil Perusahaan	4

1.7.3	Bab III Tinjauan Pustaka	4
1.7.4	Bab IV Perancangan Infrastruktur Sistem	4
1.7.5	Bab V Implementasi Sistem	4
1.7.6	Bab VI Pengujian dan Evaluasi	4
1.7.7	Bab VII Kesimpulan dan Saran	4
BAB II PROFIL PERUSAHAAN		5
2.1.	Profil Dinas Pendidikan Provinsi Jawa Timur	5
2.2.	Lokasi	5
2.3.	Struktur Organisasi	6
2.4.	Deskripsi Kegiatan	6
BAB III TINJAUAN PUSTAKA		9
3.1.	Jenkins	9
3.2.	Docker	9
3.3.	Sonar	9
3.4.	Lumen	10
3.5.	Postgresql	10
3.6.	Web Server (Nginx)	10
BAB IV PERANCANGAN INFRASTRUKTUR SISTEM		13
4.1.	Perancangan Infrastruktur Sistem	13
4.1.1.	Desain Sistem	13
BAB V IMPLEMENTASI SISTEM		17
5.1.	Flowchart Sistem	17
5.2.	Implementasi Container Service	18

5.2.1. Konfigurasi Dockerfile	18
Kode Sumber 5.1. Konfigurasi pada Dockerfile	20
5.2.2. Konfigurasi docker-compose	20
Kode Sumber 5.2. Konfigurasi docker-compose	25
5.3. Implementasi Container Database	25
5.3.1. Konfigurasi docker-compose	25
Kode Sumber 5.3. Konfigurasi docker-compose	26
Kode Sumber 5.4. Konfigurasi docker-compose Sonar	28
5.5. Implementasi Jenkins	28
Kode Sumber 5.5. Konfigurasi Nginx	30
BAB VI PENGUJIAN DAN EVALUASI	31
6.1. Tujuan Pengujian	31
6.2. Kriteria Pengujian	31
6.3. Skenario Pengujian	31
6.4. Evaluasi Pengujian	32
BAB VII KESIMPULAN DAN SARAN	33
7.1. Kesimpulan	33
7.2. Saran	33
DAFTAR PUSTAKA	35
BIODATA PENULIS I	37

DAFTAR GAMBAR

Gambar 2.1. Struktur Organisasi.....	6
Gambar 4.1. Desain Arsitektur Server.....	13
Gambar 4.1. Flowchart Sistem.....	18

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Spesifikasi Server.....	14
Tabel 6.1 Hasil Evaluasi Pengujian.....	32

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Implementasi Docker Untuk Pengelolaan Aplikasi
SIKAD PT Sentra Vidya Utama (SEVIMA)**

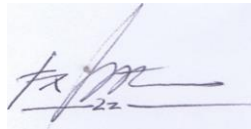
Oleh:

Restu Agung Parama

05111840000123

Disetujui oleh Pembimbing Kerja Praktik:

1. Ir. F.X. Arunanto, M.Sc.
NIP. 197007141997031002



(Pembimbing Departemen)

2. Anandre Forastero, S.Psi.,
M.Psi., Psikolog



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Implementasi Docker Untuk Pengelolaan Aplikasi PT Sentra Vidya Utama (SEVIMA)

Nama Mahasiswa : Restu Agung Parama
NRP : 05111840000123
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Ir. F.X. Arunanto, M.Sc.
Pembimbing Lapangan : Anandre Forastero, S.Psi., M.Psi.,
Psikolog

ABSTRAK

PT Sentra Vidya Utama merupakan perusahaan konsultan dan pengembang teknologi informasi yang memiliki bidang fokus dalam sektor Edukasi dan sektor Pemerintahan . Produk yang saya kerjakan saat melakukan Kerja Praktik adalah SIAKAD, yaitu sistem informasi akademik untuk membantu para mahasiswa dan kampus dalam melaksanakan perkuliahan. Pengguna utama aplikasi yang saya kembangkan adalah mahasiswa dan dosen.

Aplikasi dibuat dengan menggunakan framework Lumen dan memiliki beberapa service, beberapa diantaranya adalah autentikasi, perkuliahan, dan nilai. Saya diminta untuk menyiapkan arsitektur yang tepat agar sistem dapat berjalan dengan kendala seminimal mungkin. Sistem ini menggunakan metode container dengan bantuan webserver NGINX, Docker, dan database PostgreSQL.

Kata Kunci : Container, SIAKAD, NGINX, PostgreSQL

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT atas penyertaan dan karunia-Nya sehingga penulis dapat menyelesaikan salah satu kewajiban penulis sebagai mahasiswa Departemen Teknik Informatika ITS yaitu Kerja Praktik yang berjudul: Implementasi Docker Untuk Pengelolaan Aplikasi PT Sentra Vidya Utama (SEVIMA).

Penulis menyadari bahwa masih banyak kekurangan baik dalam melaksanakan kerja praktik maupun penyusunan buku laporan kerja praktik ini. Namun penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi.

Melalui buku laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu menyusun laporan kerja praktik baik secara langsung maupun tidak langsung antara lain:

1. Kedua orang tua penulis.
2. Bapak Ir. F.X. ARUNANTO, M.Sc. selaku dosen pembimbing kerja praktik.
3. Bapak Anandre Forastero, S.Psi., M.Psi., Psikolog selaku pembimbing lapangan selama kerja praktik berlangsung.
4. Teman-teman penulis yang senantiasa memberikan semangat ketika penulis melaksanakan KP.

Surabaya, 20 April 2022
Restu Agung Parama

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini dunia telah berkembang menjadi era digital. Semua layanan dapat disajikan secara online dengan adanya teknologi yang dapat dimanfaatkan untuk mempermudah manusia dalam melakukan layanan tersebut. Bila dibandingkan dengan yang dulu, kita membaca koran atau majalah dengan membeli fisik koran atau majalah. Kini, untuk membaca koran kita dapat membuka website atau aplikasi penyedia koran untuk membacanya. Website atau aplikasi tidak hanya terbatas untuk membaca koran. Dengan website atau aplikasi manusia seakan-akan dapat melakukan segala hal dalam satu tempat. Banyak sekali fungsi dari website atau aplikasi, sebagai contohnya pembayaran pajak, e-banking, pembelajaran, dan lain lain.

Setiap perguruan tinggi memiliki sistem informasi akademik untuk menunjang perkuliahan mulai dari pengambilan mata kuliah, perwalian, hingga penilaian. Ketika masa pengambilan mata kuliah berlangsung banyak mahasiswa yang mengakses sistem tersebut, sehingga beberapa fitur lain tidak bisa berjalan. Oleh karena itu, dibutuhkan suatu sistem microservice untuk mengelola proses-proses tersebut agar setiap service bisa berjalan dengan lancar tanpa dipengaruhi resource service lain. Untuk menunjang microservice ini dibutuhkan sistem terisolasi agar setiap service yang berada dalam satu server tidak saling mengganggu, sistem terisolasi ini bisa disebut container. Sistem informasi akademik yang saya kelola adalah untuk perguruan tinggi yang telah bekerja sama dengan PT Sentra Vidya Utama. Pada saat KP, saya diberi

kesempatan untuk merancang dan mengimplementasikan infrastruktur dari sistem informasi akademik tersebut.

1.2 Tujuan

Tujuan kerja praktik ini adalah menyelesaikan kewajiban nilai kerja praktik sebesar 2 sks dan membantu PT Sentra Vidya Utama (SEVIMA) untuk menyelesaikan permasalahan sistem informasi akademik dalam bentuk container microservice.

1.3 Manfaat

Manfaat yang diperoleh dengan adanya Docker adalah untuk mengelola container service atau fitur dalam SIAKAD agar setiap service bisa berjalan dengan lancar tanpa dipengaruhi sistem service lain, sehingga Ketika salah satu service mengalami kendala service lain masih bisa diakses dengan lancar.

1.4 Rumusan Masalah

Rumusan masalah dari kerja praktik ini adalah sebagai berikut:

1. Bagaimana implementasi Docker untuk pengelolaan aplikasi PT Sentra Vidya Utama (SEVIMA)?

1.5 Lokasi dan Waktu Kerja Praktik

Sehubungan dengan adanya pandemi dan diberlakukannya *Hybrid*, pengerjaan kerja praktik ini lakukan secara *remote* dan *offline* di kantor.

Adapun kerja praktik dimulai pada tanggal 1 September 2021 hingga 28 Februari 2022.

1.6 Metodologi Kerja Praktik

Metodologi dalam pembuatan buku kerja praktik meliputi :

1.6.1 Perumusan Masalah

Untuk mengetahui kebutuhan dari aplikasi , kami mengikuti rapat bersama tim developer. Sebelum rapat, saya diberi arahan oleh Pak Halim selaku jajaran C dari PT SEVIMA, yang menjelaskan mengenai konsep serta tujuan dari produk SIAKAD SEVIMA ini. Pada saat rapat kami dijelaskan bagaimana rencana microservice ini akan dibuat. Setelah dijelaskan, pemimpin tim developer merumuskan fitur - fitur apa saja yang akan dijadikan menjadi satu service. Dibahas juga perkiraan jumlah pengakses website untuk merencanakan arsitektur yang sebaiknya digunakan.

1.6.2 Studi Literatur

Setelah mendapat gambaran bagaimana sistem tersebut berjalan, kami diberitahu tinjauan apa saja yang akan diimplementasikan untuk membuat container aplikasi beroperasi. Tinjauan yang dipakai meliputi Lumen, NGINX, Docker, Jenkins, Sonarqube, Postgresql, dan lain-lain.

1.6.3 Perancangan Sistem

Setelah tinjauan yang dipakai telah diberitahu, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem.

1.6.4 Implementasi Sistem

Tahap implementasi meliputi implementasi kode penerapan container. Pada tahap ini kami Melakukan *deployment* pada aplikasi yang telah dibuat oleh tim developer.

1.6.5 Pengujian dan Evaluasi

Setelah website yang telah direncanakan telah jadi, perlu adanya evaluasi untuk menguji apakah service sesuai dengan harapan client. Jika masih belum sesuai atau perlu menambah fitur, rapat akan dilakukan lagi

untuk mem-*floor*-kan fitur - fitur apa saja yang perlu diperbaiki atau ditambah.

1.6.6 Kesimpulan dan Saran

Pengujian yang dilakukan ini telah memenuhi syarat yang diinginkan, dan berjalan dengan baik dan lancar.

1.7 Sistematika Laporan

1.7.1 Bab I Pendahuluan

Bab ini berisi latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2 Bab II Profil Perusahaan

Bab ini berisi gambaran umum PT Sentra Vidya Utama mulai dari profil, lokasi perusahaan.

1.7.3 Bab III Tinjauan Pustaka

Bab ini berisi dasar teori dari teknologi yang digunakan dalam menyelesaikan proyek kerja praktik.

1.7.4 Bab IV Perancangan Infrastruktur Sistem

Bab ini berisi mengenai tahap perancangan sistem aplikasi dalam menyelesaikan proyek kerja praktik.

1.7.5 Bab V Implementasi Sistem

Bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6 Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.7 Bab VII Kesimpulan dan Saran

Bab ini berisi kesimpulan dan saran yang didapat dari proses pelaksanaan kerja praktik.

BAB II

PROFIL PERUSAHAAN

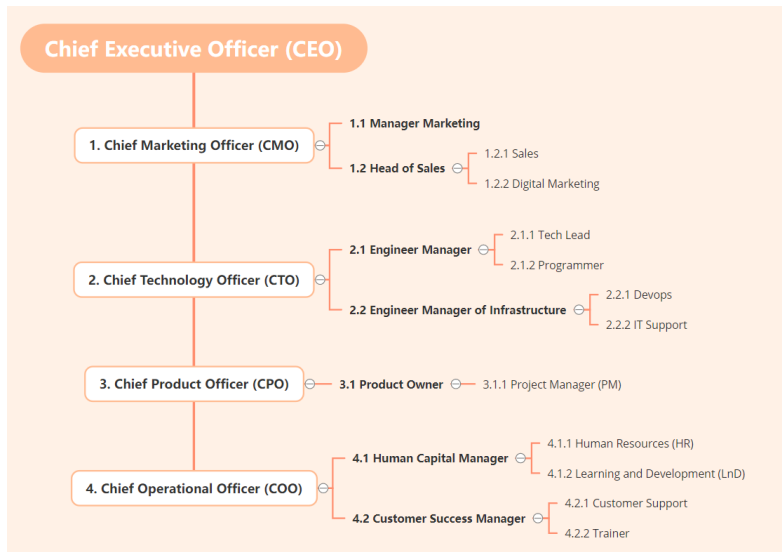
2.1. Profil Dinas Pendidikan Provinsi Jawa Timur

PT Sentra Vidya Utama (SEVIMA) merupakan perusahaan konsultan dan pengembang teknologi informasi yang didirikan pada tahun 2004. Saat ini memiliki lokasi kantor di Surabaya dengan didukung oleh tenaga ahli muda yang kompeten, inovatif dan profesional yang berpengalaman di bidangnya. SEVIMA memiliki bidang fokus dalam sektor Edukasi.

2.2. Lokasi

Jl. Medokan Asri Tengah, MA 2 Q-16, Surabaya
Jawa Timur – Indonesia

2.3. Struktur Organisasi



Gambar 2.1. Struktur Organisasi

2.4. Deskripsi Kegiatan

Kegiatan magang yang dilakukan selama enam bulan (1 September 2021 - 28 Februari 2022) pada PT Sentra Vidya Utama bagian Devops adalah sebagai berikut :

1. Docker

Membuat image docker dari beberapa aplikasi yang sudah ada dan di pecah menjadi *service-service* kecil, hal ini dilakukan untuk mempermudah proses deploy aplikasi ke server.

2. Jenkins

Mengintegrasikan repositori yang ada dengan proses deploy, sehingga ketika terdapat *commit* atau *push* baru dari repositori yang telah diintegrasikan akan membuat Jenkins menjalankan script untuk proses deploy di dalam server.

3. Sonarqube

Setiap aplikasi terbuat dari beberapa bahasa pemrograman yang disatukan, kemudian setiap ada perubahan pada aplikasi sonarqube akan mengecek apakah terdapat celah yang membuat orang lain berbuat jahat pada sistem kita. Sonarqube diintegrasikan dengan Jenkins, sehingga sebelum aplikasi di deploy akan di cek terlebih dahulu.

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

3.1. Jenkins

Jenkins adalah sebuah open source automation server untuk mengotomatisasi tugas-tugas di dalam proses continuous integration dan continuous delivery pada perangkat lunak. Jenkins merupakan aplikasi berbasis Java yang dapat dipasang dari repositori Ubuntu atau dengan mengunduh dan menjalankan file Web application ARchive (WAR), sebuah koleksi file yang sudah lengkap dan tinggal dijalankan di sebuah server.

3.2. Docker

Docker merupakan open-source platform yang digunakan untuk pengembangan, shipping, dan menjalankan aplikasi. Docker memungkinkan sebuah aplikasi dapat dipisahkan dari infrastruktur utama karena Docker dibangun berdasarkan teknologi container. Berbeda dengan virtualisasi yang harus berjalan didalam infrastruktur utamanya.

3.3. Sonar

Sonar adalah layanan analisis kode berbasis cloud yang dirancang untuk mendeteksi masalah kualitas kode untuk 25 bahasa pemrograman yang berbeda. SonarCloud akan terus memastikan pemeliharaan, reliability, dan keamanan kode. SonarCloud dapat digunakan untuk semua proyek open-source secara gratis.

3.4. Lumen

Lumen adalah sebuah web application network yang bersifat open source yang digunakan untuk membangun aplikasi php dinamis. CodeIgniter menjadi sebuah framework PHP dengan model MVC (Model, View, Controller) untuk membangun website dinamis dengan menggunakan PHP yang dapat mempercepat pengembang untuk membuat sebuah aplikasi web. Selain ringan dan cepat, Lumen juga memiliki dokumentasi yang super lengkap disertai dengan contoh implementasi kodenya dan merupakan turunan dari framework Laravel.

3.5. Postgresql

Merupakan salah satu sistem manajemen relasional basis data SQL yang bersifat open source. Postgresql digunakan untuk menyimpan data - data yang dapat saling berelasi dengan data yang lain. Untuk melakukan operasi CRUD (Create, Read, Update, Delete) memerlukan query, lalu query tersebut dikirim ke dalam database Postgresql. Pada saat query tersebut sampai pada database, database akan mengolah query tersebut dan diterapkan sesuai perintah query.

3.6. Web Server (Nginx)

Salah satu komponen penting di dalam website adalah web server. Web server berfungsi sebagai penerima request dari browser yang kemudian memberikan tanggapan dengan mengirimkan halaman situs web dalam bentuk dokumen HTML. Nginx adalah web server yang cukup populer saat ini. Selain memberikan performa yang andal, Nginx juga mempunyai beberapa fitur canggih lain yang mudah dikonfigurasi. Jadi tentu saja akan membuat website Anda lebih powerful dan canggih.

Nginx menawarkan penggunaan konkurensi tinggi yang dapat menangani ribuan koneksi secara bersamaan dengan penggunaan memori yang rendah. Nginx menggunakan proses *asynchronous* (mengeksekusi kode sebelum perintah sebelumnya selesai dijalankan) daripada membuat proses baru untuk setiap eksekusi baris kode yang ada. Tidak hanya menambahkan proses *asynchronous*, Nginx juga melakukan pendekatan *event-driven*. Pendekatan *event-driven* menangani *request* (permintaan atau proses) di dalam satu thread saja. Jadi Nginx tidak membuat proses baru ketika ada permintaan (*web request*), tapi ditangani di dalam satu thread (*asynchronous* dan pendekatan *event-driven*). Melalui berbagai macam teknologi ini Nginx merevolusi bagaimana web server bekerja dan berhasil menjadi web server paling populer sampai dengan saat ini.

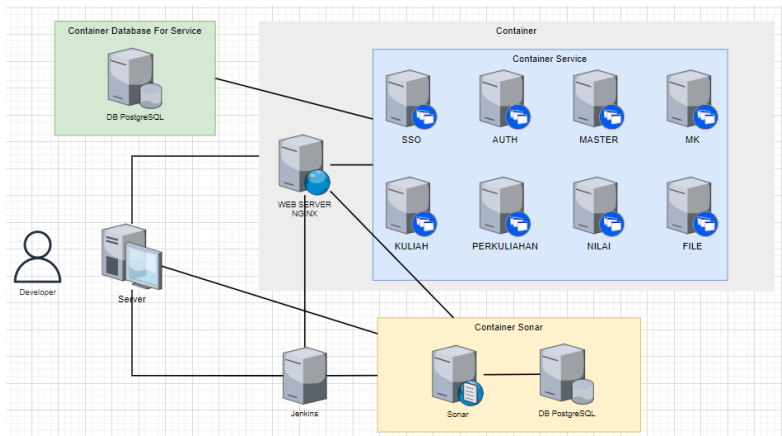
[Halaman ini sengaja dikosongkan]

BAB IV PERANCANGAN INFRASTRUKTUR SISTEM

4.1. Perancangan Infrastruktur Sistem

4.1.1. Desain Sistem

Desain arsitektur pada Sistem Informasi Akademik ini dilakukan pembagian pengelompokkan container yang dibuat. Terdapat 8 buah service SIAKAD yang akan digunakan dan web service Nginx sebagai mengatur pembagian service dan app yang akan di expose keluar. Kemudian dibuat 2 buah database yang akan digunakan oleh beberapa service berbeda. Desain server tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1. Desain Arsitektur Server

Gambar 4.1 merupakan desain arsitektur server yang digunakan pada Sistem Informasi Akademik. Satu container berupa nginx sebagai gateway bertugas meneruskan request ke aplikasi Jenkins, Sonar, atau service SIAKAD.

Pada container DB yang digunakan untuk service SIAKAD dibuat tersendiri untuk mencegah terjadinya down aplikasi lainnya ketika service SIAKAD mengalami masalah. Kemudian container Sonar berisi database serta aplikasi sonar itu sendiri yang berguna untuk menjalankan testing kepada service-service ketika ada update perubahan pada repositori. Sonar akan berjalan ketika dipanggil oleh aplikasi Jenkins. Lalu terdapat aplikasi Jenkins yang digunakan untuk menjalankan pipeline.

Berikut detail semua pada server yang dapat dilihat pada tabel 4.1.

Tabel 4.1. Spesifikasi Server

No	Nama Service	Fungsi	Spesifikasi Server
1.	Server	Perangkat Utama Server	8 VCPU, 8 GB RAM, 50 GB DISK
2.	DB PostgreSQL (Hijau)	DB untuk service SIAKAD	-
3.	DB PostgreSQL (Kuning)	DB untuk aplikasi Sonar	-
4.	Jenkins	Aplikasi untuk menjalankan pipeline	-
5.	Sonar	Aplikasi untuk menjalankan testing	-
6.	Web Server NGINX	Sebagai web server dan gateway semua service dan aplikasi	-
7.	SSO	Service SIAKAD untuk	-

		gateway utama service lainnya	
8.	AUTH	Service SIAKAD untuk autentikasi	-
9.	MASTER	Service SIAKAD untuk semua detail tambahan yang tidak ada di service lainnya	-
10.	MK	Service SIAKAD untuk mata kuliah	-
11.	KULIAH	Service SIAKAD untuk semua urusan kuliah seperti data dosen mahasiswa dan lainnya	-
12.	PERKULIAHAN	Service SIAKAD untuk masa perkuliahan seperti kelas, dosen wali, krs, dan lainnya.	-
13.	NILAI	Service SIAKAD untuk semua nilai	-
14.	FILE	Service SIAKAD untuk file-file eksternal	-

[Halaman ini sengaja dikosongkan]

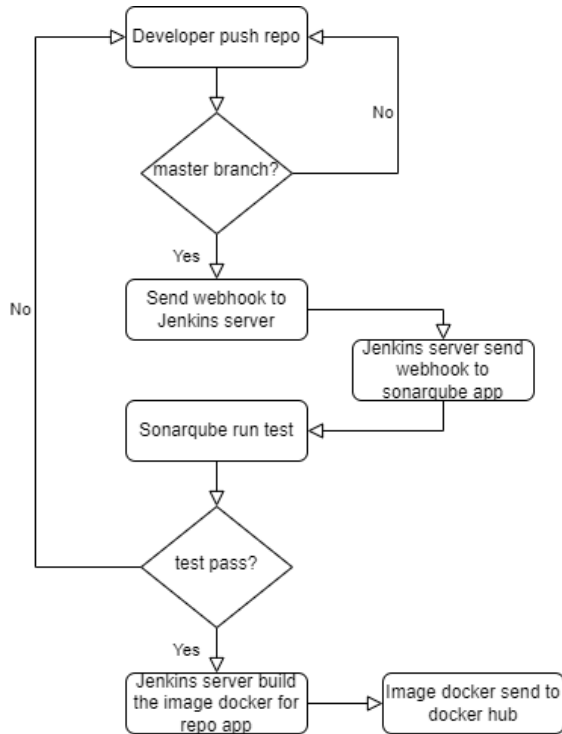
BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari sistem yang saya buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian flowchart sistem, implementasi container service, implementasi container database, implementasi container sonar, dan implementasi jenkins.

5.1. Flowchart Sistem

Dalam proses pembuatan docker image dimulai dari *developer* melakukan perubahan pada repositori dan melakukan *push commit*. Kemudian Jenkins server akan menjalankan test dengan bantuan aplikasi sonarqube, apabila hasilnya lulus Jenkins akan meneruskan proses *build image* docker dari repositori yang di-*push*, namun apabila hasil tesnya gagal maka proses Jenkins akan berhenti yang menandakan ada kesalahan dalam repo-nya, oleh karena itu developer perlu memperbaiki dan melakukan *push* ulang. Berikut merupakan flowchart dari sistem ini.



Gambar 5.1. Flowchart Sistem

5.2. Implementasi Container Service

Implementasi service php ini berfokus pada pembuatan image docker dari service yang sudah dibuat tim developer. Adapun implementasinya sebagai berikut :

5.2.1. Konfigurasi Dockerfile

Pada implementasi container service ini, Dockerfile berfungsi sebagai OS yang menjadi dasar dari service-service yang akan dibuat imagenya. Konfigurasi Dockerfile akan ditampilkan pada Kode Sumber 5.1:

```

FROM php:8.0-fpm-alpine

# Arguments defined in docker-compose.yml
ARG user
ARG uid

RUN apk update && apk add\
    git \
    curl \
    libpng-dev \
    libxml2-dev \
    zip \
    unzip \
    libmcrypt-dev \
    libmemcached-dev \
    imagemagick-dev

# Clear cache
RUN rm -rf /var/lib/apt/lists/*
# add bash
RUN apk add --no-cache --upgrade bash
# install pgsql
RUN set -ex \
    && apk --no-cache add \
    postgresql-dev

# Install PHP extensions
RUN apk add --no-cache \
    $PHPIZE_DEPS \
    openssl-dev \
    oniguruma-dev

RUN apk update && apk add --no-cache libpq \
    && docker-php-ext-configure pgsql -with-
pgsql=/usr/local/pgsql \
    && docker-php-ext-install pdo pdo_pgsql pgsql

RUN pecl install mcrypt redis memcached imagick
RUN docker-php-ext-enable mcrypt redis memcached imagick

```

```

RUN docker-php-ext-install mbstring exif pcntl bcmath gd xml

# Get latest Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Create system user to run Composer and Artisan Commands
RUN adduser -D -G www-data -u $uid -h /home/$user $user
RUN addgroup $user root
RUN mkdir -p /home/$user/.composer && \
    chown -R $user:www-data /home/$user && \
    chown -R $user:www-data /var/www

# Set working directory
WORKDIR /var/www

```

Kode Sumber 5.1. Konfigurasi pada Dockerfile

5.2.2. Konfigurasi docker-compose

Konfigurasi docker-compose ini dilakukan untuk membuat image docker. Konfigurasi tersebut akan ditampilkan pada Kode Sumber 5.2 berikut:

```

version: '3.4'

x-environment: &localvar
  HOST_IP: "172.31.44.222"
  DB_USERNAME: "siakad"
  DB_PASSWORD: "password"

services:
  sso:
    build:
      args:
        user: sevima
        uid: 1000
      context: ./
      dockerfile: Dockerfile
    image: siakad
    ports:

```

```
- 9001:9000
container_name: sso
environment:
  <<: *localvar
  DB_DATABASE: "siakad_sso"
  SERVICES_NAME: "SSO"
working_dir: /var/www/sso
volumes:
  - ./sso:/var/www/sso
  - ../env.prod:/var/www/sso/.env
networks:
  - siakad-cloud

auth:
  image: siakad
  ports:
    - 9002:9000
  # image: siakad-auth
  container_name: auth
  environment:
    <<: *localvar
    DB_DATABASE: "siakad_auth"
    SERVICES_NAME: "AUTH"
  working_dir: /var/www/auth
  volumes:
    - ./auth:/var/www/auth
    - ../env.prod:/var/www/auth/.env
  networks:
    - siakad-cloud

master:
  image: siakad
  ports:
    - 9003:9000
  # image: siakad-master
  container_name: master
  environment:
    <<: *localvar
    DB_DATABASE: "siakad_master"
    SERVICES_NAME: "MASTER"
  working_dir: /var/www/master
```

```

volumes:
  - ./master:/var/www/master
  - ./env.prod:/var/www/master/.env
networks:
  - siacad-cloud

mk:
  image: siacad
  ports:
    - 9004:9000
  # image: siacad-mk
  container_name: mk
  environment:
    <<: *localvar
    DB_DATABASE: "siacad_mk"
    SERVICES_NAME: "MK"
  working_dir: /var/www/mk
  volumes:
    - ./mk:/var/www/mk
    - ./env.prod:/var/www/mk/.env
  networks:
    - siacad-cloud

kuliah:
  image: siacad
  ports:
    - 9005:9000
  # image: siacad-kuliah
  container_name: kuliah
  environment:
    <<: *localvar
    DB_DATABASE: "siacad_kuliah"
    SERVICES_NAME: "KULIAH"
  working_dir: /var/www/kuliah
  volumes:
    - ./kuliah:/var/www/kuliah
    - ./env.prod:/var/www/kuliah/.env
  networks:
    - siacad-cloud

perkuliahan:

```

```
image: siakad
ports:
  - 9006:9000
# image: siakad-perkuliahan
container_name: perkuliahan
environment:
  <<: *localvar
  DB_DATABASE: "siakad_perkuliahan"
  SERVICES_NAME: "PERKULIAHAN"
working_dir: /var/www/perkuliahan
volumes:
  - ./perkuliahan:/var/www/perkuliahan
  - ./env.prod:/var/www/perkuliahan/.env
networks:
  - siakad-cloud

nilai:
image: siakad
ports:
  - 9007:9000
# image: siakad-nilai
container_name: nilai
environment:
  <<: *localvar
  DB_DATABASE: "siakad_nilai"
  SERVICES_NAME: "NILAI"
working_dir: /var/www/nilai
volumes:
  - ./nilai:/var/www/nilai
  - ./env.prod:/var/www/nilai/.env
networks:
  - siakad-cloud

file:
image: siakad
ports:
  - 9008:9000
# image: siakad-file
container_name: file
environment:
  <<: *localvar
```

```
DB_DATABASE: "siakad_file"
SERVICES_NAME: "FILE"
working_dir: /var/www/file
volumes:
  - ./file:/var/www/file
  - ./env.prod:/var/www/file/.env
networks:
  - siakad-cloud

nginx:
  image: nginx
  container_name: nginx
  restart: unless-stopped
  ports:
    - 127.0.0.1:8000:8001
    - 127.0.0.1:8001:8002
    - 127.0.0.1:8002:8003
    - 127.0.0.1:8003:8004
    - 127.0.0.1:8004:8005
    - 127.0.0.1:8005:8006
    - 127.0.0.1:8006:8007
    - 127.0.0.1:8007:8008
  volumes:
    - ./sso:/var/www/sso
    - ./auth:/var/www/auth
    - ./master:/var/www/master
    - ./mk:/var/www/mk
    - ./kuliah:/var/www/kuliah
    - ./perkuliahan:/var/www/perkuliahan
    - ./nilai:/var/www/nilai
    - ./file:/var/www/file
    - ./docker-compose/nginx:/etc/nginx/conf.d/
    - ./logs/nginx:/var/log/nginx/
  networks:
    - siakad-cloud

networks:
  siakad-cloud:
    external:
      name: db_siakad-cloud
```


5.3. Implementasi Container Database

Implementasi container database ini berfokus pada pembuatan container database yang dilakukan secara terpisah dengan container service lainnya. Adapun implementasi container database sebagai berikut:

5.3.1. Konfigurasi docker-compose

Konfigurasi docker-compose ini dilakukan untuk membuat image docker. Konfigurasi tersebut akan ditampilkan pada Kode Sumber 5.3 berikut:

```
version: "3.7"
services:
  postgres:
    image: postgres
    container_name: siakad-db
    command: ["postgres", "-c",
"config_file=/etc/postgresql.conf"]
    volumes:
      - ./PGDATA/pg_hba.conf:/etc/pg_hba.conf
      - ./PGDATA/postgresql.conf:/etc/postgresql.conf
      - ./docker-postgresql-multiple-databases:/docker-
entrypoint-initdb.d
    environment:
      -
      POSTGRES_MULTIPLE_DATABASES=siakad_sso,siakad_auth,siakad_master
,siakad_mk,siakad_kuliah,siakad_perkuliahan,siakad_nilai,siakad_
file
      - POSTGRES_USER=siakad
      - POSTGRES_PASSWORD=password
    ports:
      - "5434:5432"
    networks:
      - siakad-cloud

redis:
```

```

image: redis
container_name: siacad-redis
ports:
  - "63790:6379"
networks:
  - siacad-cloud

networks:
  siacad-cloud:
    driver: bridge

```

Kode Sumber 5.3. Konfigurasi docker-compose

5.4. Implementasi Container Sonar

Implementasi container sonar ini berfokus pada konfigurasi docker-compose untuk membuat image sonar. Adapun detail konfigurasinya akan ditampilkan pada Kode Sumber 5.4 berikut:

```

version: "3.4"

services:
  sonarqube:
    image: sonarqube
    container_name: sonar
    restart: unless-stopped
    environment:
      - SONARQUBE_JDBC_USERNAME=sonar
      - SONARQUBE_JDBC_PASSWORD=sonarpass
      - SONARQUBE_JDBC_URL=jdbc:postgresql://db:5432/sonarqube
    ports:
      - "9090:9090"
      - "9092:9092"
    volumes:
      - sonarqube_conf:/opt/sonarqube/conf
      - sonarqube_data:/opt/sonarqube/data
      - sonarqube_extensions:/opt/sonarqube/extensions
      - sonarqube_bundled-plugins:/opt/sonarqube/lib/bundled-plugins

```

```

networks:
  - sonar-virtual

db:
  image: postgres
  container_name: sonar-db
  restart: unless-stopped
  # command: ["postgres", "-c",
"config_file=/etc/postgresql.conf"]
  volumes:
    - ./pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf
  - ./postgresql.conf:/var/lib/postgresql/data/postgresql.conf
    - sonarqube_db:/var/lib/postgresql
      # This needs explicit mapping due to
https://github.com/docker-library/postgres/blob/4e48e3228a30763913ece952c611e5e9b95c8759/Dockerfile.template#L52
    - postgresql_data:/var/lib/postgresql/data
  environment:
    - POSTGRES_USER=sonar
    - POSTGRES_PASSWORD=sonarpass
    - POSTGRES_DB=sonarqube
    # - POSTGRES_HOST_AUTH_METHOD=scram-sha-256
    # - POSTGRES_INITDB_ARGS=--auth-host=scram-sha-256
  networks:
    - sonar-virtual

volumes:
  postgresql_data:
  sonarqube_bundled-plugins:
  sonarqube_conf:
  sonarqube_data:
  sonarqube_db:
  sonarqube_extensions:
networks:
  sonar-virtual:
    driver: bridge

```

5.5. Implementasi Jenkins

Implementasi Jenkins ini berfokus pada konfigurasi nginx untuk membuat Jenkins dan container sonar dapat diakses dalam 1 port. Adapun detail konfigurasinya akan ditampilkan pada Kode Sumber 5.5 berikut:

```
# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    # SSL configuration
    #
    # listen 443 ssl default_server;
    # listen [::]:443 ssl default_server;
    #
    # Note: You should disable gzip for SSL traffic.
    # See: https://bugs.debian.org/773332
    #
    # Read up on ssl_ciphers to ensure a secure
configuration.
    # See: https://bugs.debian.org/765782
    #
    # Self signed certs generated by the ssl-cert package
    # Don't use them in a production server!
    #
    # include snippets/snakeoil.conf;

    #root /var/www/html;

    # Add index.php to the list if you are using PHP
    index index.php index.html index.htm index.nginx-
debian.html;

    server_name _;
```

```

# ProxyPass /sonar!
# alias /sonar http://127.0.0.1:9090
# proxy_set_header Host $host;
# proxy_set_header X-Forwarded-For $remote_addr;

# location /login {
#     rewrite ^/login(.*)$ /jenkins/login$1;
# }
location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a
404.
    # #try_files $uri $uri/ =404;
    # rewrite    ^/jenkins/(.*)$ /$1 break;
    proxy_pass  http://127.0.0.1:8080;
    # if ($request_body ~* ^(.*)/sonar) {
    #     proxy_pass
http://127.0.0.1:9090;
    # }

}

location /sonar {
    # rewrite    ^/sonar/(.*)$ /$1 break;
# proxy_pass  http://127.0.0.1:8080/$1;
    proxy_pass  http://127.0.0.1:9090/sonar;
    # proxy_set_header Host $host;
}

# pass PHP scripts to FastCGI server
#
#location ~ /\.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/run/php/php7.4-fpm.sock;
#    # With php-cgi (or other tcp sockets):
#    fastcgi_pass 127.0.0.1:9000;
#}

# deny access to .htaccess files, if Apache's document

```

```
root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}
}
```

Kode Sumber 5.5. Konfigurasi Nginx

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba terhadap Aplikasi Sistem Akademik (SIAKAD) PT SEVIMA. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan fitur-fitur yang sudah dirancang.

6.1. Tujuan Pengujian

Pengujian dilakukan terhadap Aplikasi Sistem Akademik (SIAKAD) PT SEVIMA guna menguji kemampuan arsitektur dalam melayani permintaan sistem aplikasi.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan berikut :

- a. Kemampuan arsitektur untuk melayani service.
- b. Kemampuan arsitektur untuk melayani query data dari aplikasi ke database.
- c. Kemampuan arsitektur untuk menyimpan data yang diinput melalui aplikasi.
- d. Kemampuan arsitektur untuk integrasi antar service.

6.3. Skenario Pengujian

Skenario pengujian dilakukan dengan melakukan peran sebagai user yang akan menjalankan fitur-fitur. Langkah-langkah untuk setiap kebutuhan fungsionalitas yaitu sebagai berikut :

1. Pengguna bisa membuka service.
2. Pengguna bisa membuat pipeline dari Jenkins.
3. Pengguna bisa mengakses sonarqube.

4. Pengguna bisa mengintegrasikan sonarqube dengan Jenkins.

6.4. Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi SIAKAD terhadap kasus skenario uji coba. Tabel 6.1 di bawah ini menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

Tabel 6.1. Hasil Evaluasi Pengujian

Kriteria Pengujian	Hasil Pengujian
Server dapat melayani service	Terpenuhi
Server dapat menyediakan data yang diminta aplikasi dari database	Terpenuhi
Server dapat menyimpan data yang di-input user ke database	Terpenuhi
Server dapat mengintegrasikan service	Terpenuhi

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan perancangan arsitektur sistem aplikasi SIAKAD pada kegiatan kerja praktek di PT Sentra Vidya Utama adalah sebagai berikut :

- a. Arsitektur sistem yang dibangun telah sesuai dengan permintaan.
- b. Dengan adanya aplikasi SIAKAD yang telah dibuat microservice dan dikontainerisasi, Tim developer dapat dengan mudah memonitoring dan mengelola data untuk memaksimalkan potensi server.

7.2. Saran

Saran untuk perancangan arsitektur sistem aplikasi PPDB adalah sebagai berikut :

- a. Perlu adanya pembagian database untuk setiap service agar tidak bermasalah ketika salah satu service mengalami down.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Niagahoster Blog. 2020. Apa Itu Nginx dan Cara Kerjanya - Niagahoster Blog. [ONLINE] Available at: <https://www.niagahoster.co.id/blog/nginx-adalah/>. [Diakses 07 01 2022].
- [2] "What is Jenkins?" [Online]. Tersedia: <https://www.edureka.co/blog/what-is-jenkins>. [Diakses 06 01 2022].
- [3] Chacon, Scott, dan Ben Straub. 2021. "Pro Git" (edisi ke-2). Mountain View: Apress. Tersedia: <https://git-scm.com/book/en/v2>.
- [4] Brown, A. dan Wilson, G. 2012. "The Architecture of Open Source Application". Lulu Press, Inc.
- [5] "Docker overview" [Online]. Tersedia: <https://docs.docker.com/get-started/overview/>. [Diakses 07 01 2022].
- [6] "Using Sentry — Error Monitoring with Ionic Angular Applications". [Online]. Tersedia: <https://medium.com/enappd/using-sentry-error-tracking-with-ionic-application-d206277a3416>. [Diakses 07 01 2022].
- [7] "Sonar Cloud Docs". [Online]. Tersedia: <https://docs.sonarcloud.io/>. [Diakses 07 01 2022].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS I

Nama : Restu Agung Parama
Tempat, Tanggal Lahir : Bekasi, 22 Januari 2001
Jenis Kelamin : Laki-laki
Telepon : +6285771699404
Email : restu.ap2001@gmail.com

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2018
Semester : 8 (Delapan)