



## **KERJA PRAKTIK – IF184801**

### **Perancangan dan Implementasi Infrastruktur Sistem Penerimaan Peserta Didik Baru (PPDB) SD dan SMP Kota Surabaya**

Dinas Pendidikan Kota Surabaya

Jalan Jagir Wonokromo No.354-356, Sidosermo,  
Wonokromo, Sidosermo, Wonocolo, Kota SBY, Jawa Timur  
60244

Periode: 20 Februari 2020 - 5 Agustus 2020

#### **Oleh:**

Ismail Arifin 05111740007007

Ardy Wahyu Setyawan 05111840000050

#### **Pembimbing Jurusan**

Yudhi Purwananto, S.Kom., M.Kom.

#### **Pembimbing Lapangan**

Dwi Sunaryono, S.Kom., M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2021

*[Halaman ini sengaja dikosongkan]*



*[Halaman ini sengaja dikosongkan]*

LEMBAR PENGESAHAN

**KERJA PRAKTIK**  
**Perancangan dan Implementasi Infrastruktur Sistem**  
**Penerimaan Peserta Didik Baru (PPDB) SD dan SMP Kota**  
**Surabaya**

Oleh:

Ismail Arifin

05111740007007

Ardy Wahyu Setyawan

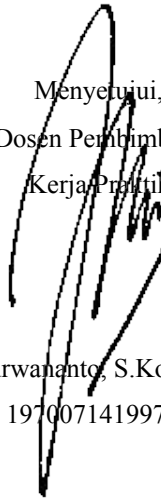
05111840000050

Mengetahui,  
Pembimbing Lapangan  
Kerja Praktik



Dwi Sunaryono, S.Kom., M.Kom.  
NIP. 197205281997021001

Menyetujui,  
Dosen Pembimbing  
Kerja Praktik



Yudhi Purwananto, S.Kom., M.Kom.  
NIP. 197007141997031002

SURABAYA  
JANUARI 2021

*[Halaman ini sengaja dikosongkan]*

**Perancangan dan Implementasi Infrastruktur Sistem  
Penerimaan Peserta Didik Baru (PPDB) SD dan SMP Kota  
Surabaya**

Nama Mahasiswa : Ismail Arifin  
NRP : 05111740007007  
Nama Mahasiswa : Ardy Wahyu Setyawan  
NRP : 05111840000050  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Jurusan : Yudhi Purwananto, S.Kom., M.Kom.  
Pembimbing Lapangan : Dwi Sunaryono, S.Kom., M.Kom

**ABSTRAK**

Dinas Pendidikan Kota Surabaya merupakan instansi pemerintah yang bertanggung jawab tentang hal yang berkaitan dengan pendidikan di Kota Surabaya. Produk yang kami kerjakan saat melakukan Kerja Praktik adalah PPDB, yaitu sistem penerimaan siswa baru yang membantu calon peserta didik baru untuk memilih SD atau SMP tujuan.

Aplikasi dibuat dengan menggunakan framework Laravel dan memiliki beberapa fitur, beberapa diantaranya adalah pendaftaran peserta didik, informasi pelaksanaan PPDB, dan perangkingan. Kami diminta untuk menyiapkan arsitektur yang tepat agar sistem dapat diakses oleh masyarakat dengan kendala seminimal mungkin. Sistem ini menggunakan metode load balancing dengan bantuan webserver NGINX, HAProxy, dan database MySQL.

***Kata Kunci : Website, PPDB, NGINX, HAProxy***

*[Halaman ini sengaja dikosongkan]*



## **KATA PENGANTAR**

Puji syukur kami haturkan kepada Allah SWT karena berkat rahmat-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Informatika, yakni Kerja Praktik (KP).

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik secara langsung, maupun tidak langsung dalam pelaksanaan kerja praktik hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Bapak Yudhi Purwananto, S.Kom., M.Kom., selaku dosen pembimbing kerja praktik.
3. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp., Ph.D selaku koordinator Kerja Praktik.
4. Bapak Dwi Sunaryono, S.Kom., M.Kom., selaku pembimbing lapangan kami di Departemen Informatika ITS

Surabaya, 8 Januari 2021  
Ismail Arifin, Ardy Wahyu Setyawan

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvii
DAFTAR KODE	xix
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. Rumusan Masalah	3
1.6.2. Studi Literatur	3
1.6.3. Analisis dan Perancangan Sistem	3
1.6.4. Implementasi Sistem	4
1.6.5. Pengujian dan Evaluasi	4
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	5
1.7.3.	Bab III Tinjauan Pustaka	5
1.7.4.	Bab IV Analisis dan Perancangan Sistem	5
1.7.5.	Bab V Implementasi Sistem	5
1.7.6.	Bab VI Pengujian dan Evaluasi	5
1.7.7.	Bab VII Kesimpulan dan Saran	6
BAB II PROFIL PERUSAHAAN		7
2.1	Sejarah Dinas Pendidikan Kota Surabaya	7
2.2	Visi dan Misi Dinas Pendidikan Kota Surabaya	7
2.3	Tujuan Dinas Pendidikan Kota Surabaya	8
BAB III TINJAUAN PUSTAKA		9
3.1.	Server	9
3.2.	Nginx	9
3.3.	PHP-FPM	9
3.4.	MySQL	10
3.5.	Git	10
3.6.	Cloudflare	10
3.7.	HTML	11
3.8.	Javascript	11
3.9.	HAProxy	11
BAB IV ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM		13
4.1.	Analisis Sistem	13

4.2.	Perancangan Infrastruktur Sistem	13
<b>BAB V IMPLEMENTASI SISTEM</b>		<b>19</b>
5.1	Implementasi Load Balancing NGINX	19
5.1.1.	Konfigurasi NGINX	19
5.1.2.	Konfigurasi php-fpm	22
5.2	Implementasi Load Balancing Database	Hasil
Ranking		32
5.2.1.	Konfigurasi HAProxy	32
5.2.2.	Konfigurasi Database Output	33
5.3.	Implementasi Database Master	36
5.4.	Implementasi Database Slave	40
<b>BAB VI PENGUJIAN DAN EVALUASI</b>		<b>45</b>
6.1	Tujuan Pengujian	45
6.2	Kriteria Pengujian	45
6.3	Skenario Pengujian	45
6.4	Evaluasi Pengujian	46
<b>BAB VII KESIMPULAN DAN SARAN</b>		<b>47</b>
7.1.	Kesimpulan	47
7.2.	Saran	47
<b>DAFTAR PUSTAKA</b>		<b>49</b>
<b>BIODATA PENULIS I</b>		<b>51</b>
<b>BIODATA PENULIS II</b>		<b>51</b>

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR TABEL**

Tabel 4.1. Spesifikasi setiap server	14
Tabel 4.2. Spesifikasi software setiap server	16
Tabel 6.1. Evaluasi pengujian	46

*[Halaman ini sengaja dikosongkan]*



## **DAFTAR GAMBAR**

Gambar 4.1 Desain Arsitektur Server

14

*[Halaman ini sengaja dikosongkan]*

## **DAFTAR KODE**

Kode Sumber 5.1 Konfigurasi pada nginx.conf	19
Kode Sumber 5.2 Konfigurasi default website PPDB Surabaya pada NGINX	20
Kode Sumber 5.3 Konfigurasi php-fpm PPDB Surabaya SMP	22
Kode Sumber 5.4 Konfigurasi php-fpm PPDB Surabaya SD	25
Kode Sumber 5.5 Konfigurasi php-fpm PPDB Surabaya Dapur	28
Kode Sumber 5.6 Konfigurasi HAProxy	33
Kode Sumber 5.7 Konfigurasi mysqld.cnf pada server database output	33
Kode Sumber 5.8 Konfigurasi mysqld.cnf pada server database master	36
Kode Sumber 5.9 Konfigurasi mysqld.cnf pada server database slave	40

*[Halaman ini sengaja dikosongkan]*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi informasi saat ini sudah tidak dapat terlepas dari teknologi internet. Dengan adanya teknologi internet semua layanan dapat disajikan secara online yang memberi dampak kemudahan dalam melaksanakan aktivitas sehari-hari. Salah satu bidang yang terkena dampak yaitu pendidikan. Teknologi telah membantu bidang pendidikan dalam pertumbuhan pembelajaran bergerak dan pembelajaran jarak jauh.

Penerimaan Peserta Didik Baru (PPDB) untuk sekolah negeri merupakan kegiatan rutin setiap tahun pada bidang pendidikan di Indonesia yang dilaksanakan di setiap daerah pada tingkat SD, SMP dan SMA. Salah satu kota yang mengadakan penerimaan peserta didik baru adalah Kota Surabaya. Untuk mempermudah dalam melakukan pendaftaran ke sekolah tujuan, dibutuhkan suatu sistem untuk mengelola penerimaan peserta didik baru tersebut. Terutama dalam kondisi pandemi seperti sekarang ini, semua proses yang biasanya masih dapat dilakukan secara tatap muka harus dilakukan secara online. Penerimaan peserta didik baru yang kami kelola adalah untuk jenjang SD dan SMP. Pada saat KP, kami diberi kesempatan untuk merancang dan mengimplementasikan infrastruktur dari sistem penerimaan peserta didik baru tersebut.

## **1.2. Tujuan**

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk membuat infrastruktur aplikasi PPDB Surabaya 2020 agar aplikasi tersebut dapat diakses dengan lancar.

## **1.3. Manfaat**

Manfaat dari pembuatan infrastruktur aplikasi ini yaitu untuk membuat aplikasi PPDB Surabaya 2020 dapat diakses dengan lancar.

## **1.4. Rumusan Masalah**

Adapun rumusan masalah pada kerja praktik pembuatan infrastruktur aplikasi untuk sistem penerimaan peserta didik baru (PPDB) SMP dan SD Surabaya adalah bagaimana merancang arsitektur server agar aplikasi PPDB Surabaya 2020 dapat diakses dengan lancar.

## **1.5. Lokasi dan Waktu Kerja Praktik**

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut :

Lokasi : Departemen Informatika ITS

Alamat : Jalan Teknik Kimia ITS, Sukolilo, Surabaya

Waktu : 20 Februari 2020 - 5 Agustus 2020

Kerja praktik dilaksanakan secara daring dikarenakan pandemi COVID-19.

## **1.6. Metodologi Kerja Praktik**

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

### **1.6.1 Perumusan Masalah**

Untuk mengetahui kebutuhan dari website, kami mengikuti rapat bersama tim developer dan Dosen. Pada saat rapat kami dijelaskan bagaimana konsep dan proses penerimaan peserta didik baru. Setelah dijelaskan, pemimpin tim developer merumuskan fitur - fitur apa saja yang akan diterapkan pada website yang akan dibuat. Dibahas juga perkiraan jumlah pengakses website untuk merencanakan arsitektur yang sebaiknya digunakan.

### **1.6.2 Studi Literatur**

Pada tahap ini setelah mendapat gambaran bagaimana sistem tersebut akan berjalan. Tinjauan yang akan dipakai meliputi NGINX, HAProxy, MySQL, dan lain-lain. Setelah itu dilakukan studi literatur lanjut mengenai cara implementasinya.

### **1.6.3 Analisis dan Perancangan Sistem**

Setelah melakukan tinjauan pustaka, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem. Pada website ini tim *developer* setuju

untuk menggunakan arsitektur desain MVC (Model - View - Controller).

#### **1.6.4 Implementasi Sistem**

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami melakukan deployment pada aplikasi yang telah dibuat oleh tim developer.

#### **1.6.5 Pengujian dan Evaluasi**

Pengujian dilakukan dengan menguji fitur-fitur yang ada oleh client. Jika terdapat fitur yang belum sesuai atau tidak bekerja sebagaimana mestinya maka akan dilakukan rapat untuk melakukan evaluasi terhadap fitur tersebut.

#### **1.6.6 Kesimpulan dan Saran**

Pengujian yang dilakukan ini telah sesuai syarat yang diinginkan dan fitur-fitur aplikasi berjalan dengan baik dan lancar.

### **1.7. Sistematika Laporan**

Laporan kerja praktik ini terdiri dari enam bab dengan rincian sebagai berikut:

#### **1.7.1 Bab I Pendahuluan**

Bab ini berisi tentang latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu



pelaksanaan, metodologi, serta sistematika laporan pengerjaan kerja praktik.

### **1.7.2 Bab II Profil Perusahaan**

Bab ini berisi tentang profil perusahaan tempat kami melaksanakan kerja praktik, yakni Dinas Pendidikan Kota Surabaya.

### **1.7.3 Bab III Tinjauan Pustaka**

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di Dinas Pendidikan Kota Surabaya.

### **1.7.4 Bab IV Analisis dan Perancangan Sistem**

Bab ini berisi analisis sistem aplikasi dalam mengerjakan kerja praktik.

### **1.7.5 Bab V Implementasi Sistem**

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

### **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**

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

## **BAB II**

### **PROFIL PERUSAHAAN**

#### **2.1 Profil Dinas Pendidikan Kota Surabaya**

Dinas Pendidikan Kota Surabaya merupakan instansi pemerintah yang bertanggung jawab tentang hal yang berkaitan dengan pendidikan di Kota Surabaya. Bertugas melaksanakan urusan pemerintahan Kota Surabaya bidang pendidikan serta melaksanakan tugas-tugas lain berkaitan dengan pendidikan yang diberikan oleh Walikota sesuai dengan bidang tugasnya.

#### **2.2 Visi dan Misi Dinas Pendidikan Kota Surabaya**

**Visi :**

“MENJADIKAN SURABAYA SEBAGAI BAROMETER DAN INSPIRATOR PENDIDIKAN NASIONAL”

**Misi :**

1. Mewujudkan layanan akses pendidikan yang bermutu secara merata dan berkeadilan.
2. Memperkuat budaya lokal dalam sendi - sendi pendidikan.

### **2.3 Tujuan Dinas Pendidikan Kota Surabaya**

Tujuan dari Dinas Pendidikan Kota Surabaya adalah sebagai berikut.

1. Mewujudkan pemerataan aksesibilitas dan kualitas pendidikan formal.
2. Mewujudkan pemerataan aksesibilitas dan kualitas pendidikan non formal.
3. Meningkatkan perlindungan dan pengembangan budaya lokal.
4. Mewujudkan peningkatan minat dan budaya baca masyarakat melalui peningkatan akses baca.

## **BAB III**

### **TINJAUAN PUSTAKA**

#### **3.1. Server**

Server atau dalam bahasa Indonesia biasa disebut peladen merupakan suatu sistem komputer yang memiliki layanan khusus berupa penyimpanan data. Data yang disimpan melalui server berupa informasi dan beragam jenis dokumen yang kompleks. Layanan tersebut ditujukan khusus untuk client yang berkebutuhan dalam menyediakan informasi untuk pengguna atau pengunjungnya.

#### **3.2. Nginx**

NGINX adalah software web server yang open source. Ketika pertama kali dirilis, NGINX hanya berfungsi sebagai HTTP web serving saja. Namun sekarang, software tersebut juga berperan sebagai reverse proxy, HTTP load balancer, dan email proxy untuk IMAP, POP3, dan SMTP.

#### **3.3. PHP-FPM**

PHP-FPM (FastCGI Process Manager) adalah implementasi FastCGI alternatif untuk bahasa PHP. PHP-FPM merupakan interpreter PHP yang terpisah dari aplikasi web server. Setiap request atas script PHP yang masuk ke web server akan diteruskan (forward) ke FastCGI melalui socket atau koneksi TCP/IP.

### **3.4. MySQL**

MySQL merupakan database engine atau server database yang mendukung bahasa database pencarian SQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread, multi-user.

### **3.5. Git**

Git adalah salah satu *tool* yang sering digunakan dalam proyek pengembangan software. Git adalah salah satu sistem pengontrol versi (Version Control System) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds.

Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri. Git dikenal juga dengan distributed revision control (VCS terdistribusi), artinya penyimpanan database Git tidak hanya berada dalam satu tempat saja. Semua orang yang terlibat dalam pengkodean proyek akan menyimpan database Git, sehingga akan memudahkan dalam mengelola proyek baik online maupun offline.

### **3.6. Cloudflare**

CloudFlare adalah jaringan pengiriman konten yang bertindak sebagai proxy antara website dan visitor. Melayani cache konten statis dari lokasi terdekat (CloudFlare memiliki banyak server yang tersebar di beberapa negara), hal itu menjadikan loading sebuah website lebih cepat. Sebagai gambaran sederhana, CloudFlare bertindak sebagai penghubung antara pengunjung dan server website. Tujuannya

supaya website terhindar dari serangan berbahaya yang dilakukan oleh pengunjung website.

### **3.7. HTML**

Bahasa standar internasional yang digunakan untuk membuat halaman web. HTML menggambarkan struktur dan isi semantik dari sebuah dokumen. HTML biasanya digabungkan dengan css dan javascript. css untuk memperindah tampilan dan javascript untuk client side scripting language.

### **3.8. Javascript**

Javascript adalah sebuah bahasa tingkat tinggi yang dinamis. Javascript memiliki banyak sekali fungsionalitas seperti web application, backend, desktop application, internet of things (IoT), dan lain - lain. Pada buku kerja praktik ini javascript digunakan untuk client side scripting language yang tertanam pada HTML sebuah website. Javascript juga memiliki banyak library yang dapat digunakan contohnya nodejs, axiosjs, bluebirdjs, vuejs, angularjs, reactjs, animatejs, dan lain - lain.

### **3.9. HAProxy**

HAProxy adalah sebuah software open source yang menyediakan load balancer dengan ketersediaan 12 tinggi dan proxy server untuk aplikasi berbasis TCP dan HTTP yang tersebar pada banyak server. HAProxy membantu membagi beban request database ke beberapa server berbeda untuk menghindari terjadinya kelebihan request pada database.

*[Halaman ini sengaja dikosongkan]*



## **BAB IV**

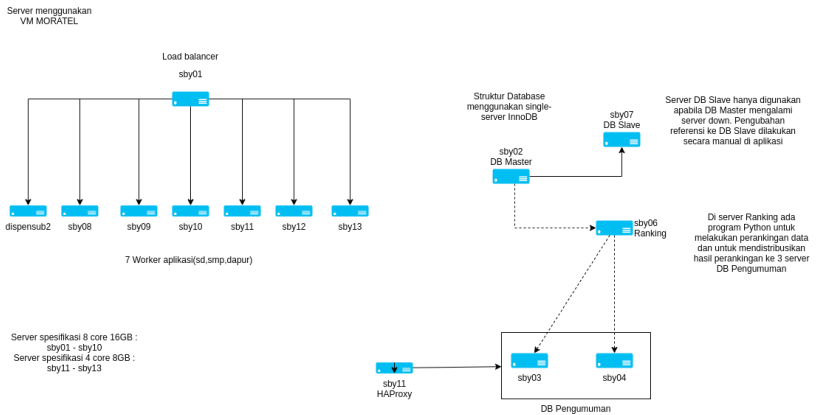
# **ANALISIS DAN PERANCANGAN INFRASTRUKTUR SISTEM**

### **4.1. Analisis Sistem**

Secara umum, aplikasi PPDB merupakan sistem penerimaan peserta didik baru berbasis website dan mobile yang digunakan dalam pendaftaran calon peserta didik baru. Aplikasi PPDB surabaya terdiri atas 3 aplikasi yaitu SD, SMP dan Dapur.

### **4.2. Perancangan Infrastruktur Sistem**

Desain arsitektur pada Sistem Informasi PPDB SD dan SMP Kota Surabaya ini dilakukan pembagian beban untuk web server load balancing dengan menggunakan NGINX, yaitu menggunakan satu server sebagai server gateway load balancing, 7 server sebagai web server aplikasi(SD, SMP, dan Dapur). Selain itu, terdapat beberapa server lain yang berperan sebagai server DB master, DB ranking, DB output, penyimpanan static. Desain server clustering tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1. Desain Arsitektur Server

Gambar 4.1. merupakan desain arsitektur server clustering yang digunakan pada Sistem Informasi PPDB SD dan SMP Kota Surabaya 2020.

Spesifikasi setiap server dapat dilihat pada tabel 4.1 dibawah ini.

Tabel 4.1. Spesifikasi setiap server

No	Nama Server	Fungsi	Spesifikasi Server
1	sby01	Web Server, Load Balancer PHP	8 Core 16GB RAM
2	sby02	DB Master	8 Core 16GB RAM
3	sby03	DB Output	8 Core 16GB RAM

4	sby04	DB Output	8 Core 16GB RAM
5	sby05	Penyimpanan file	8 Core 16GB RAM
5	sby06	DB Ranking	8 Core 16GB RAM
6	sby07	DB Slave	8 Core 16GB RAM
7	sby08	Worker PHP	8 Core 16GB RAM
8	sby09	Worker PHP	8 Core 16GB RAM
9	sby10	Worker PHP	8 Core 16GB RAM
10	sby11	Worker PHP, Penyimpanan static, Load Balancer DB Output	4 Core 8GB RAM
11	sby12	Worker PHP	4 Core 8GB RAM
12	sby13	Worker PHP	4 Core 8GB RAM
13	dispensub2	Worker PHP	4 Core 4GB RAM

Spesifikasi software setiap server dapat dilihat pada tabel 4.2 dibawah ini.

Tabel 4.2. Spesifikasi software setiap server

No	Nama Server	Fungsi	Spesifikasi Software
1	sby01	Web Server, Load Balancer PHP	NGINX, PHP
2	sby02	DB Master	MySQL
3	sby03	DB Output	MySQL
4	sby04	DB Output	MySQL
5	sby05	Penyimpanan file	NGINX, MySQL, php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-pspell, php7.2-xml, php7.2-fpm
6	sby06	DB Ranking	MySQL
7	sby07	DB Slave	MySQL
8	sby08	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-pspell,

			php7.2-xml, php7.2-fpm
9	sby09	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-openssl, php7.2-xml, php7.2-fpm
10	sby10	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-openssl, php7.2-xml, php7.2-fpm
11	sby11	Worker PHP, Penyimpanan static, Load Balancer DB Output	HAProxy, NGINX,
12	sby12	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-openssl, php7.2-xml, php7.2-fpm
13	sby13	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring,

			php7.2-mcrypt, php7.2-mysql, php7.2-openssl, php7.2-xml, php7.2-fpm
14	dispensub2	Worker PHP	php7.2-cli, php7.2-cgi, php7.2-curl, php7.2-gd, php7.2-mbstring, php7.2-mcrypt, php7.2-mysql, php7.2-openssl, php7.2-xml, php7.2-fpm

## **BAB V**

### **IMPLEMENTASI SISTEM**

Bab ini membahas tentang implementasi dari sistem yang kami buat. Implementasi ini akan dibagi ke dalam beberapa bagian, yaitu bagian implementasi load balancing PHP, implementasi load balancing database, implementasi database utama, implementasi database slave.

#### **5.1. Implementasi Load Balancing NGINX**

Implementasi load balancing php ini berfokus pada pembagian beban request situs web pada server-server php. Adapun implementasi load balancing php sebagai berikut :

##### **5.1.1. Konfigurasi NGINX**

Pada implementasi load balancing php ini, NGINX berlaku sebagai web server dan load balancer. Konfigurasi default NGINX untuk situs website PPDB Surabaya SD, SMP, Dapur akan ditampilkan pada Kode Sumber 5.1 dan 5.2 berikut :

```
user www-data;
worker_processes 8;
pid /run/nginx.pid;
worker_rlimit_nofile 100000;

events {
    worker_connections 100000;
    multi_accept on;
}

http {

    ##
    # Basic Settings
```

```

    ##

    ##
    # SSL Settings
    ##

    ##
    # Virtual Host Configs
    ##

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}

```

Kode Sumber 5.1 Konfigurasi pada nginx.conf

```

# Default server configuration

upstream ppdbscopy {
    ##
    # Ip dari worker-worker php
    ##
}

server {
    listen 80;

    ##
    # Konfigurasi SSL
    ##
}

```



```

index index.html index.htm index.php;

if ($http_x_forwarded_proto = "http") {
    return 301 https://$host/$request_uri;
}

##
# Servername untuk ppdb surabaya
##

##
# Directory tempat penyimpanan file aplikasi
##

# Konfigurasi untuk keseluruhan website
location / {
    # auth_basic "Mohon maaf. Server sedang
dipersiapkan";
    # auth_basic_user_file /etc/nginx/.htpasswd;
    try_files $uri $uri/ /index.php;
}

location ~ \.php$ {
    ##
    # Konfigurasi php-fpm
    ##
}
}

```

Kode Sumber 5.2 Konfigurasi default website PPDB Surabaya pada NGINX

### 5.1.2. Konfigurasi php-fpm

Konfigurasi php-fpm ini dilakukan pada setiap worker php dari situs web ppdb surabaya. Konfigurasi setiap website berbeda untuk aplikasi SD, SMP dan Dapur. Konfigurasi tersebut akan ditampilkan pada Kode Sumber 5.3, 5.4, 5.5 berikut :

```
; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be
replaced by the
; pool name ('www' here)
[www]

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default
user's group
;     will be used.
user = www-data
group = www-data

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4
address on
;                 a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific
IPv6 address on
;                 a specific port;
; 'port'           - to listen on a TCP socket to all addresses
;                 (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
```

```

; Note: This value is mandatory.
listen = 9000

; Set permissions for unix socket, if one is used. In Linux,
read/write
; permissions must be set in order to allow connections from a
web server. Many
; BSD-derived systems allow connections regardless of
permissions. The owner
; and group can be specified either by name or by their numeric
IDs.
; Default Values: user and group are set as the running user
; mode is set to 0660
listen.owner = www-data
listen.group = www-data

; Choose how the process manager will control the number of
child processes.
; Possible Values:
; static - a fixed number (pm.max_children) of child processes;
; dynamic - the number of child processes are set dynamically
based on the
; following directives. With this process management,
there will be
; always at least 1 children.
; pm.max_children - the maximum number of children
that can
; be alive at the same time.
; pm.start_servers - the number of children created on
startup.
; pm.min_spare_servers - the minimum number of children
in 'idle'
; state (waiting to process). If the number
of 'idle' processes is less than this

```

```

;          number then some children will be created.
;          pm.max_spare_servers - the maximum number of
children in 'idle'
;          state (waiting to process). If the number
;          of 'idle' processes is greater than this
;          number then some children will be killed.
; ondemand - no children are created at startup. Children will be
forked when
;          new requests will connect. The following parameter are
used:
;          pm.max_children      - the maximum number of
children that
;          can be alive at the same time.
;          pm.process_idle_timeout - The number of seconds after
which
;          an idle process will be killed.
; Note: This value is mandatory.
pm = dynamic

; The number of child processes to be created when pm is set to
'static' and the
; maximum number of child processes when pm is set to 'dynamic'
or 'ondemand'.
; This value sets the limit on the number of simultaneous requests
that will be
; served. Equivalent to the ApacheMaxClients directive with
mpm_prefork.
; Equivalent to the PHP_FCGI_CHILDREN environment variable
in the original PHP
; CGI. The below defaults are based on a server without much
resources. Don't
; forget to tweak pm.* to fit your needs.
; Note: Used when pm is set to 'static', 'dynamic' or 'ondemand'
; Note: This value is mandatory.

```

```

pm.max_children = 200

; The number of child processes created on startup.
; Note: Used only when pm is set to 'dynamic'
; Default Value: min_spare_servers + (max_spare_servers -
min_spare_servers) / 2
pm.start_servers = 8

; The desired minimum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.min_spare_servers = 4

; The desired maximum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.max_spare_servers = 8

```

Kode Sumber 5.3 Konfigurasi php-fpm PPDB Surabaya SMP

```

; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be
replaced by the
; pool name ('www' here)
[sd]

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default
user's group
;   will be used.
user = www-data
group = www-data

```

```

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4
address on
;
; a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific
IPv6 address on
;
; a specific port;
; 'port' - to listen on a TCP socket to all addresses
;
; (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = 9001

; Set permissions for unix socket, if one is used. In Linux,
read/write
; permissions must be set in order to allow connections from a
web server. Many
; BSD-derived systems allow connections regardless of
permissions.
; Default Values: user and group are set as the running user
;
; mode is set to 0660
listen.owner = www-data
listen.group = www-data

; Choose how the process manager will control the number of
child processes.
; Possible Values:
; static - a fixed number (pm.max_children) of child processes;
; dynamic - the number of child processes are set dynamically
based on the
;
; following directives. With this process management,
there will be

```

```

; always at least 1 children.
; pm.max_children - the maximum number of children
that can
; be alive at the same time.
; pm.start_servers - the number of children created on
startup.
; pm.min_spare_servers - the minimum number of children
in 'idle'
; state (waiting to process). If the number
; of 'idle' processes is less than this
; number then some children will be created.
; pm.max_spare_servers - the maximum number of
children in 'idle'
; state (waiting to process). If the number
; of 'idle' processes is greater than this
; number then some children will be killed.
; ondemand - no children are created at startup. Children will be
forked when
; new requests will connect. The following parameter are
used:
; pm.max_children - the maximum number of
children that
; can be alive at the same time.
; pm.process_idle_timeout - The number of seconds after
which
; an idle process will be killed.
; Note: This value is mandatory.
pm = dynamic

; The number of child processes to be created when pm is set to
'static' and the
; maximum number of child processes when pm is set to 'dynamic'
or 'ondemand'.
; This value sets the limit on the number of simultaneous requests

```

```

that will be
; served. Equivalent to the ApacheMaxClients directive with
mpm_prefork.
; Equivalent to the PHP_FCGI_CHILDREN environment variable
in the original PHP
; CGI. The below defaults are based on a server without much
resources. Don't
; forget to tweak pm.* to fit your needs.
; Note: Used when pm is set to 'static', 'dynamic' or 'ondemand'
; Note: This value is mandatory.
pm.max_children = 400
; The number of child processes created on startup.
; Note: Used only when pm is set to 'dynamic'
; Default Value: min_spare_servers + (max_spare_servers -
min_spare_servers) / 2
pm.start_servers = 8

; The desired minimum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.min_spare_servers = 4

; The desired maximum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.max_spare_servers = 8

```

Kode Sumber 5.4 Konfigurasi php-fpm PPDB Surabaya SD

```

; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be
replaced by the

```



```

; pool name ('www' here)
[dapur]

; Unix user/group of processes
; Note: The user is mandatory. If the group is not set, the default
user's group
;   will be used.
user = www-data
group = www-data

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
; 'ip.add.re.ss:port' - to listen on a TCP socket to a specific IPv4
address on
;           a specific port;
; '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific
IPv6 address on
;           a specific port;
; 'port' - to listen on a TCP socket to all addresses
;           (IPv6 and IPv4-mapped) on a specific port;
; '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = 9002

; Set permissions for unix socket, if one is used. In Linux,
read/write
; permissions must be set in order to allow connections from a
web server. Many
; BSD-derived systems allow connections regardless of
permissions.
; Default Values: user and group are set as the running user
;           mode is set to 0660
listen.owner = www-data
listen.group = www-data

```

```

; Choose how the process manager will control the number of
child processes.
; Possible Values:
; static - a fixed number (pm.max_children) of child processes;
; dynamic - the number of child processes are set dynamically
based on the
;         following directives. With this process management,
there will be
;         always at least 1 children.
;         pm.max_children    - the maximum number of children
that can
;                             be alive at the same time.
;         pm.start_servers   - the number of children created on
startup.
;         pm.min_spare_servers - the minimum number of children
in 'idle'
;                             state (waiting to process). If the number
;                             of 'idle' processes is less than this
;                             number then some children will be created.
;         pm.max_spare_servers - the maximum number of
children in 'idle'
;                             state (waiting to process). If the number
;                             of 'idle' processes is greater than this
;                             number then some children will be killed.
; ondemand - no children are created at startup. Children will be
forked when
;         new requests will connect. The following parameter are
used:
;         pm.max_children    - the maximum number of
children that
;                             can be alive at the same time.
;         pm.process_idle_timeout - The number of seconds after
which

```

```

;                                     an idle process will be killed.
; Note: This value is mandatory.
pm = dynamic

; The number of child processes to be created when pm is set to
'static' and the
; maximum number of child processes when pm is set to 'dynamic'
or 'ondemand'.
; This value sets the limit on the number of simultaneous requests
that will be
; served. Equivalent to the ApacheMaxClients directive with
mpm_prefork.
; Equivalent to the PHP_FCGI_CHILDREN environment variable
in the original PHP
; CGI. The below defaults are based on a server without much
resources. Don't
; forget to tweak pm.* to fit your needs.
; Note: Used when pm is set to 'static', 'dynamic' or 'ondemand'
; Note: This value is mandatory.
pm.max_children = 100

; The number of child processes created on startup.
; Note: Used only when pm is set to 'dynamic'
; Default Value: min_spare_servers + (max_spare_servers -
min_spare_servers) / 2
pm.start_servers = 2

; The desired minimum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'
; Note: Mandatory when pm is set to 'dynamic'
pm.min_spare_servers = 1

; The desired maximum number of idle server processes.
; Note: Used only when pm is set to 'dynamic'

```

```
; Note: Mandatory when pm is set to 'dynamic'  
pm.max_spare_servers = 3
```

Kode Sumber 5.5 Konfigurasi php-fpm PPDB Surabaya Dapur

## 5.2. Implementasi Load Balancing Database Hasil Ranking

Implementasi load balancing database ini berfokus pada pembagian beban request hasil ranking pada dua server database dengan bantuan HAProxy. Adapun implementasi load balancing database hasil ranking sebagai berikut :

### 5.2.1. Konfigurasi HAProxy

Pada implementasi load balancing database ini, HAProxy berlaku sebagai load balancer. Konfigurasi HAProxy untuk situs web PPDB SD dan SMP Kota Surabaya akan ditampilkan pada Kode Sumber 5.6 berikut :

```
global  
    log 127.0.0.1 local0  
    log 127.0.0.1 local1 notice  
    maxconn 10240  
    nbproc 4  
    cpu-map 1 0  
    cpu-map 2 1  
    cpu-map 3 2  
    cpu-map 4 3  
    uid 99  
    gid 99  
    daemon  
  
defaults  
    log global
```

```

mode http
option tcplog
option dontlognull
retries 3
option redispatch
maxconn 10240

listen mysql
bind 0.0.0.0:3306
mode tcp
balance leastconn
option mysql-check user haproxy
log global
timeout client 3m
timeout server 3m
timeout connect 1m
server db03 192.168.10.7:3306 check
server db04 192.168.10.5:3306 check

```

Kode Sumber 5.6 Konfigurasi HAProxy

### 5.2.2. Konfigurasi Database Output

Konfigurasi database MySQL ini berada pada server sby03, sby04 sebagai server database output atau hasil ranking. Berikut detail konfigurasi pada Kode Sumber 5.7 :

```

[mysqld_safe]
socket      = /var/run/mysqld/mysqld.sock
nice       = 0

```

```

[mysqld]
#
# * Basic Settings
#
user          = mysql
pid-file      = /var/run/mysqld/mysqld.pid
socket        = /var/run/mysqld/mysqld.sock
port          = 3306
basedir       = /usr
datadir       = /var/lib/mysql
tmpdir        = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address      = #ip HAProxy
#
# * Fine Tuning
#
key_buffer_size      = 128M
max_allowed_packet   = 128M
thread_stack         = 192K
#thread_cache_size   = 8
thread_cache_size    = 32
# This replaces the startup script and checks MyISAM tables if
# needed
# the first time they are touched
myisam-recover-options = BACKUP
max_connections      = 2048
#table_cache         = 64
#thread_concurrency  = 10
#
# * Query Cache Configuration

```

```

#
query_cache_limit      = 20M
query_cache_size      = 384M
query_cache_type       = 1

#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file      = /var/log/mysql/mysql.log
#general_log           = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries      = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for
replication.
# note: if you are setting up a replication slave, see
README.Debian about
# other settings you may need to change.
server-id              = 2
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days      = 2
max_binlog_size       = 100M
#binlog_do_db         = test_database
#binlog_ignore_db     = include_database_name

```

```

#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are
many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI
"tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem
slave_net_timeout      = 15
innodb_flush_log_at_trx_commit      = 2
innodb_flush_method    = O_DIRECT
innodb_buffer_pool_size = 2G

```

Kode Sumber 5.7 Konfigurasi mysqld.cnf pada server database output

### 5.3. Implementasi Database Master

Implementasi database ini berfokus pada konfigurasi satu database agar mampu menerima banyak request read dan write dalam satu waktu. Adapun detail konfigurasi database utama Sistem Informasi PPDB SD dan SMP Kota Surabaya akan ditampilkan pada Kode Sumber 5.8 berikut :



```

[mysqld_safe]
socket      = /var/run/mysqld/mysqld.sock
nice        = 0

[mysqld]
#
# * Basic Settings
#
user        = mysql
pid-file    = /var/run/mysqld/mysqld.pid
socket      = /var/run/mysqld/mysqld.sock
port        = 3306
basedir     = /usr
datadir     = /var/lib/mysql
tmpdir      = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address      = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size      = 128M
max_allowed_packet   = 128M
thread_stack         = 192K
#thread_cache_size   = 8
thread_cache_size     = 32
# This replaces the startup script and checks MyISAM tables if
# needed
# the first time they are touched
mysisam-recover-options = BACKUP

```

```

#max_connections      = 2048
max_connections      = 4096

#table_cache         = 64
#thread_concurrency  = 10
#
# * Query Cache Configuration
#
query_cache_type      = 0
#query_cache_limit    = 20M
#query_cache_size     = 128M
query_cache_size      = 0
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file     = /var/log/mysql/mysql.log
#general_log          = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries     = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for
# replication.
# note: if you are setting up a replication slave, see
README.Debian about

```

```

#   other settings you may need to change.
server-id                = 1
log_bin                  = /var/log/mysql/mysql-bin.log
expire_logs_days        = 2
max_binlog_size          = 100M
#binlog_do_db            = test_database
#binlog_ignore_db        = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are
many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI
"tinyc".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem
slave_net_timeout        = 15
innodb_flush_log_at_trx_commit = 2
innodb_flush_method      = O_DIRECT

#innodb_buffer_pool_size = 4G
#innodb_log_file_size = 512M

innodb_buffer_pool_size = 8G

```

```
innodb_log_file_size = 1G

table_open_cache = 8192

#interactive_timeout = 120
#wait_timeout = 120
```

Kode Sumber 5.8 Konfigurasi mysqld.cnf pada server database master

#### 5.4. Implementasi Database Slave

Server Database Slave hanya digunakan apabila Database Master mengalami server down. Perubahan referensi ke Database Slave dilakukan secara manual di aplikasi.

```
[mysqld_safe]
socket      = /var/run/mysqld/mysqld.sock
nice       = 0

[mysqld]
#
# * Basic Settings
#
user       = mysql
pid-file   = /var/run/mysqld/mysqld.pid
socket     = /var/run/mysqld/mysqld.sock
port      = 3306
basedir    = /usr
datadir    = /var/lib/mysql
tmpdir     = /tmp
lc-messages-dir = /usr/share/mysql
skip-external-locking
```

```

#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address          = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size       = 128M
max_allowed_packet    = 128M
thread_stack          = 192K
thread_cache_size     = 8
# This replaces the startup script and checks MyISAM tables if
# needed
# the first time they are touched
myisam-recover-options = BACKUP
max_connections       = 4096
#table_cache          = 64
#thread_concurrency   = 10
#
# * Query Cache Configuration
#
query_cache_limit     = 20M
query_cache_size      = 128M
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
#general_log_file     = /var/log/mysql/mysql.log
#general_log          = 1
#
# Error log - should be very few entries.
#

```

```

log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#log_slow_queries      = /var/log/mysql/mysql-slow.log
#long_query_time = 2
#log-queries-not-using-indexes
#
# The following can be used as easy to replay backup logs or for
replication.
# note: if you are setting up a replication slave, see
README.Debian about
#   other settings you may need to change.
server-id              = 2
relay-log              = /var/log/mysql/mysql-relay-bin.log
log_bin                = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size       = 100M
#binlog_do_db          = test_database
#binlog_ignore_db      = include_database_name
#
#   IGNORING TABLE FOR REPLICATION IN SLAVE
replicate-ignore-table = ppdb_sby_sd_2018.ci_sessions
replicate-ignore-table = ppdb_sby_2018.ci_sessions
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in
/var/lib/mysql/.
# Read the manual for more InnoDB related options. There are
many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!

```

```
# chroot = /var/lib/mysql/  
#  
# For generating SSL certificates I recommend the OpenSSL GUI  
"tinyca".  
#  
# ssl-ca=/etc/mysql/cacert.pem  
# ssl-cert=/etc/mysql/server-cert.pem  
# ssl-key=/etc/mysql/server-key.pem  
slave_net_timeout      = 15  
slave-skip-errors      = 1062  
innodb_flush_log_at_trx_commit    = 2  
innodb_flush_method    = O_DIRECT
```

Kode Sumber 5.9 Konfigurasi mysqld.cnf pada server database slave

*[Halaman ini sengaja dikosongkan]*



## **BAB VI**

### **PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba terhadap Aplikasi Sistem Penerimaan Peserta Didik Baru (PPDB) SD dan SMP Kota Surabaya. Pengujian dilakukan untuk memastikan fungsionalitas dan kesesuaian hasil implementasi arsitektur dengan analisis dan perancangan arsitektur.

#### **6.1. Tujuan Pengujian**

Pengujian dilakukan terhadap Aplikasi Sistem Penerimaan Peserta Didik Baru (PPDB) SD dan SMP Kota Surabaya 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 tampilan aplikasi.
- 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 melayani upload file dari aplikasi.
- e. Kemampuan arsitektur untuk melakukan perangkingan.

#### **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. Calon Peserta Didik Baru dapat membuka aplikasi PPDB.
2. Calon Peserta Didik Baru dapat memeriksa data diri.
3. Calon Peserta Didik dapat melakukan pendaftaran ke Sekolah tujuan.
4. Calon Peserta Didik Baru dapat melihat hasil perangkingan sementara beberapa saat setelah form pendaftaran di-submit.
5. Calon Peserta Didik Baru dapat melihat jadwal, pagu, dan informasi penting terkait PPDB.

#### 6.4. Evaluasi Pengujian

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

Tabel 6.1. Evaluasi pengujian

<b>Kriteria Pengujian</b>	<b>Hasil Pengujian</b>
Server dapat melayani tampilan aplikasi	Terpenuhi
Server dapat menyediakan data yang diminta aplikasi dari database	Terpenuhi
Server dapat menyimpan data yang diinput user ke database	Terpenuhi
Server dapat menyimpan file yang diupload user	Terpenuhi
Server mampu melakukan perangkingan secara realtime setiap 5 menit	Terpenuhi

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan perancangan arsitektur sistem aplikasi PPDB adalah sebagai berikut :

- a. Arsitektur sistem yang dibangun telah sesuai dengan permintaan.
- b. Dengan adanya aplikasi ini dapat mempermudah siswa dalam melakukan pendaftaran ke sekolah tujuan dan mempermudah Dinas Pendidikan Kota Surabaya dalam memonitoring dan mengelola data siswa SD dan SMP di Kota Surabaya.

#### **7.2. Saran**

Saran untuk perancangan arsitektur sistem aplikasi PPDB adalah pada halaman-halaman website yang sering diakses dan memberatkan database sebaiknya dilakukan mekanisme caching.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] Dharma, Alfian. 2020. Apa itu Server ? Berikut Pengertian dan Fungsi Server. [online] Available at: <https://www.dicoding.com/blog/apa-itu-server/> [Diakses Desember 2020].
- [2] Ariata. 2019. Apa Itu NGINX? Dan Bagaimana Cara Kerjanya?. [online] Available at: <https://www.hostinger.co.id/tutorial/apa-itu-nginx/> [Diakses Desember 2020].
- [3] Data, Mahendra. 2018. Instalasi NGINX dan PHP-FPM pada Ubuntu 16.04. [online] Available at: <https://medium.com/@dataq/instalasi-nginx-dan-php-fpm-pada-ubuntu-16-04-41d4a13e3148> [Diakses Desember 2020].
- [4] Yasin, K. 2018. Apa itu Cloudflare? Ini Penjelasan Lengkapnya!. [online] Available at: <https://www.niagahoster.co.id/blog/apa-itu-cloudflare/> [Diakses Desember 2020].
- [5] Edel. 2016. Pengertian MySQL. [online] Available at: <http://edel.staff.unja.ac.id/blog/artikel/Pengertian-MySQL.html>. [Diakses Desember 2020].
- [6] Muhardian, Ahmad. 2021. Apa itu Git dan Kenapa Penting bagi Programmer?. [online] Available at: <https://www.petanikode.com/git-untuk-pemula/> [Diakses Desember 2020].

*[Halaman ini sengaja dikosongkan]*

## **BIODATA PENULIS I**

Nama : Ismail Arifin  
Tempat, Tanggal Lahir : Toli-Toli, 28 Maret 1999  
Jenis Kelamin : Laki-laki  
Email : Ismail.A2803@gmail.com

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika –  
FTEIC , ITS  
Angkatan : 2017  
Semester : 7 (Tujuh)

## **BIODATA PENULIS II**

Nama : Ardy Wahyu Setyawan  
Tempat, Tanggal Lahir : Malang, 6 Juli 2000  
Jenis Kelamin : Laki-laki  
Email : wahyuardy18@gmail.com

### **AKADEMIS**

Kuliah : Departemen Teknik Informatika –  
FTEIC , ITS  
Angkatan : 2018  
Semester : 5 (Lima)

*[Halaman ini sengaja dikosongkan]*