ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI1502

RANCANG BANGUN SISTEM LOAD BALANCING MENGGUNAKAN ALGORITMA BERBASIS KONTEN DAN KONTROL KETERSEDIAAN LAYANAN

BAHRUL HALIMI
NRP 5111100014

Dosen Pembimbing
Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD
Baskoro Adi P, S.Kom, M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2015

Halaman ini sengaja dikosongkan

TUGAS AKHIR - KI1502

RANCANG BANGUN SISTEM LOAD BALANCING MENGGUNAKAN ALGORITMA BERBASIS KONTEN DAN KONTROL KETERSEDIAAN LAYANAN

BAHRUL HALIMI
NRP 5111100014

Dosen Pembimbing
Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD
Baskoro Adi P, S.Kom, M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2015

Halaman ini sengaja dikosongkan

UNDERGRADUATE THESIS - KI1502

DESIGN AND IMPLEMENTASION OF LOAD BALANCING SYSTEM WITH CONTENT-BASED ALGORITHM AND AVAILABILITY CONTROL

BAHRUL HALIMI
NRP 5111100014

Supervisor
Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD

Baskoro Adi P, S.Kom, M.Kom

Department of INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya, 2015

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM LOAD BALANCING MENGGUNAKAN ALGORITMA BERBASIS KONTEN DAN KONTROL KETERSEDIAAN LAYANAN

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur dan Jaringan Komputer
Program Studi S1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :
Bahrul Halimi
NRP: 5111100014

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD .....
NIP: 197708242006041001 (Pembimbing 1)

Baskoro Adi P, S.Kom, M.Kom .....
NIP: 197708242006041001 (Pembimbing 2)

SURABAYA
Desember 2015

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM LOAD BALANCING MENGGUNAKAN ALGORITMA BERBASIS KONTEN DAN KONTROL KETERSEDIAAN LAYANAN

Nama : BAHRUL HALIMI
NRP : 5111100014
Jurusan : Teknik Informatika FTIf
Pembimbing I : Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD
Pembimbing II : Baskoro Adi P, S.Kom, M.Kom

Abstrak

Dokumen ini merupakan dokumen contoh penggunaan templat LaTeX untuk pembuatan Buku Tugas Akhir ITS.

Kata-Kunci: LaTeX, templat, Tugas Akhir, ITS.

DESIGN AND IMPLEMENTASION OF LOAD BALANCING SYSTEM WITH CONTENT-BASED ALGORITHM AND AVAILABILITY CONTROL

Name : BAHRUL HALIMI
NRP : 5111100014
Major : Informatics FTIf
Supervisor I : Royyana Muslim Ijtihadie, S.Kom, M.Kom, PhD
Supervisor II : Baskoro Adi P, S.Kom, M.Kom

Abstract

Dokumen ini merupakan dokumen contoh penggunaan templat LaTeX untuk pembuatan Buku Tugas Akhir ITS.

Kata-Kunci: LaTeX, templat, Tugas Akhir, ITS.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Om Swastyastu

Puji syukur penulis haturkan kepada Ida Sang Hyang Widhi Wasa, Tuhan Yang Maha Esa karena atas *asungkertha wara nugraha* beliau, penulis dapat menyelesaikan sebuah dokumentasi cara pembuatan Buku Tugas Akhir Sarjana menggunakan LaTeX untuk Institut Teknologi Sepuluh Nopember, Surabaya. Dokumentasi ini diharapkan dapat membantu rekan-rekan mahasiswa S1 yang menempuh semester terakhir dengan membuat buku Tugas Akhir menggunaan sistem *typesetting* LaTeX yang terbukti handal dan lumrah digunakan di bidang penelitian sains dan teknik. Dokumentasi ini dibuat menggunakan templat yang penulis buat sendiri (pada berkas `ta-its.cls`) sehingga nantinya bisa digunakan kembali sehingga pembuatan buku bisa lebih dipermudah.

Penulis menerima kritik dan saran mengenai pengembangan templat ini agar bisa menjadi lebih baik dan bisa menjadi standar *de-facto* dan *de-jure* dalam penulisan buku TA di seluruh civitas akademika ITS. Penulis dapat dihubungi melalui surel: `initrunlevel0@gmail.com`.

Sekian dan Terima Kasih. **Om Santhi Santhi Santhi Om**

Halaman ini sengaja dikosongkan

DAFTAR ISI

DAFTAR TABEL

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dam batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1 Latar Belakang

Semakin berkembangnya internet di masyarakat membuat penggunaan aplikasi berbasis web semakin diminati. Pengguna aplikasi berbasis web ini dapat dijumpai diberbagai aktivitas harian masyarakat, diantaranya bank *online*, *e-commerce*, reservasi tempat secara *online*, bahkan pendaftaran peserta didik baru secara *online*. Hal ini membuat penyedia layanan aplikasi berbasis web harus menyediakan servis yang layak sehingga aplikasi tetap berjalan dengan baik walaupun pengguna semakin bertambah.

Terjadinya *bottleneck* (penumpukan permintaan) menjadi tantangan tersendiri ketika pengembang tidak memperhatikan sumber dayanya dan berujung pada gagalnya permintaan pengguna [1]. Muncul gagasan awal dengan penggunaan kelompok server yang akan menangani permintaan ini. Kelompok server ini akan secara bergantian melayani setiap permintaan terhadap aplikasi berbasis web ini. Dengan adanya tugas bergantian ini dibutuhkan sebuah komputer yang bertugas membagi beban kerja kelompok server. Komputer ini biasa disebut pembagi muat atau *load balance*. Sistem kerja dari *load balancer* ini menggunakan sebuah algoritma yang sudah ditanam untuk kemudian digunakan untuk memilih komputer mana yang harus melayani permintaan pengguna.

Di sisi lain sebuah aplikasi berbasis web memiliki dua jenis halaman yang mungkin di akses. Yang pertama adalah halaman berisi informasi, baik hasil *query* basis data maupun tidak, selanjutnya disebut halaman informasi dan yang kedua adalah halaman yang di-

gunakan untuk mengirimkan data ke server, dalam hal ini berupa form pengisian informasi, selanjutnya disebut halaman daftar.

Dua jenis halaman ini memiliki kebutuhan yang berbeda. Untuk halaman informasi, pengguna mengharapkan akses yang cepat sedangkan untuk halaman daftar, pengguna mengharapkan data yang dimasukkan dapat diproses dengan aman. Padahal di dalam penggunaan algoritma sebelumnya dan dengan teknologi yang ada, *load balancer* tidak dapat memisahkan dua jenis permintaan ini. Algoritma yang ada sebelumnya hanya memisahkan banyak permintaan sesuai dengan ketersediaan server melayani pengguna. Padahal ketika proses pemasukkan data di dalam halaman daftar, seharusnya bisa digunakan untuk melayani permintaan pada halaman informasi.

Muncullah gagasan lain mengenai pengelompokkan permintaan berdasarkan konten yang diinginkan oleh pengguna. Pengelompokkan ini didasarkan pada dua halaman sebelumnya, yakni halaman informasi dan halaman daftar. Tujuannya untuk mengatur penggunaan sumber daya yang digunakan. Dua kelompok server terpisah akan melayani masing-masing permintaan yang berbeda. Dengan permintaan satu tipe dalam satu kelompok server, membuat kerja server menjadi lebih terpusat dan mengurangi beban yang besar.

Berbeda dengan yang terjadi saat ini, sebuah server atau bahkan dalam sebuah kelompok server, harus melayani berbagai bentuk permintaan dari pengguna, sehingga menyebabkan beban kerja server meningkat. Bahkan waktu dalam penyelesaian suatu permintaan tidak dapat diukur dalam satuan waktu yang sama karena berbedanya bentuk permintaan pengguna.

Sementara itu di dalam kelompok server yang bekerja bergantian melayani permintaan, ada kalanya sebuah server mengalami gangguan dan sama sekali tidak dapat melayani setiap permintaan pengguna. Padahal setiap permintaan yang ada masih diteruskan oleh load balancer pada server tersebut. Tidak adanya mekanis-

me untuk memindahkan permintaan dari server mati ke server yang masih aktif membuat akses ke sebuah web menjadi tidak maksimal. Oleh karena itu dibangunlah sistem ini. Dengan adanya sistem load balancing menggunakan algoritma berbasis konten yang memisahkan antara halaman informasi dan halaman daftar diharapkan dapat meningkatkan jumlah pengguna suatu halaman web dengan banyaknya bentuk permintaan dari pengguna.

1.2 Rumusan Masalah

Berikut beberapa hal yang menjadi rumusan masalah dalam tugas akhir ini:

1. Bagaimana membagi beban kerja server berdasarkan konten permintaan pengguna ?
2. Bagaimana menentukan pengelompokkan server berdasarkan konten permintaan pengguna ?
3. Bagaimana meningkatkan jumlah pengakses pada halaman informasi dengan terpisahnya akses antara halaman informasi dan halaman daftar ?
4. Bagaimana menjaga pengguna tetap dilayani kelompok server yang tersedia hingga permintaan selesai ?

1.3 Batasan Masalah

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

1. Konten permintaan pengguna dilihat dari URL yang diakses.
2. Pendefinisian kelompok konten permintaan pengguna dilakukan manual oleh manusia.
3. Sistem pembagi beban kerja diimplementasikan untuk aplikasi berbasis web.
4. Kelompok server yang bekerja dibedakan dengan besar memori yang digunakan.

1.4 Tujuan

Tugas akhir dibuat dengan beberapa tujuan. Berikut beberapa tujuan dari pembuatan tugas akhir:

1. Mampu mengategorikan permintaan pengguna terhadap suatu web berdasarkan halaman yang diakses pengguna.
2. Mampu melayani banyaknya permintaan pengguna dengan mengandalkan pengelompokan komputer.
3. Mampu meningkatkan jumlah pengakses yang dilayani dengan berhasil oleh aplikasi dibandingkan dengan akses tanpa pemisahan jenis halaman yang diakses.

1.5 Manfaat

Dengan dibangunnya *load balancer* ini diharapkan jumlah pengakses yang mampu dilayani oleh kelompok server untuk halaman informasi menjadi lebih banyak dibandingkan dengan penggunaan algoritma dan teknologi *load balancing* yang sudah ada.

BAB 2

LANDASAN TEORI

2.1 Algoritma Berbasis Konten

Munculnya algoritma ini didasarkan pada beberapa jenis permintaan pengguna yang mengakses suatu halaman web. Sebuah server melayani berbagai jenis permintaan akan memberikan waktu balasan yang beragam pula. Hal ini akan meningkatkan beban kerja server.

Dengan adanya pemisahan permintaan pengguna berdasarkan konten, kelompok server akan melayani setiap permintaan yang memang ditujukan untuknya server tersebut. Bentuk permintaan akan selalu sama sehingga waktu untuk melayani permintaan menjadi sama dan lebih terkontrol. Beban kerja server akan lebih ringan dengan adanya pembagian beban berdasarkan algoritma ini. [1]

2.2 Node JS

Merupakan sebuah platform yang dibangun di atas Chrome's JavaScript runtime dengan teknologi V8 yang mendukung proses server yang bersifat long-running. Tidak seperti platform modern yang mengandalkan multithreading, NodeJS memilih menggunakan asynchronous I/O eventing. Karena inilah NodeJS mampu bekerja dengan konsumsi memori rendah. [2] [3]

Teknologi yang tidak memanfaatkan multi-thread ini memudahkan pengembang yang terkadang kesulitan mengatur sumberdaya yang digunakan thread. Karena tidak mungkin ada sumberdaya yang terkunci karena thread yang berjalan. Akhirnya banyak yang memanfaatkan kemampuan dasar NodeJS sebagai web server.

Dengan adanya callback untuk setiap penggunaan fungsi, memungkinkan setiap pemanggilan fungsi yang tidak menghasilkan apapun, NodeJS akan *sleep*

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 1337;

http.createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World\n');
}).listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Gambar 2.1: Contoh Penggunaan NodeJS sebagai Web Server

2.3 Angular JS

Angular JS membantu dalam pembangunan halaman HTML menjadi lebih dinamis. Merupakan sebuah kumpulan alat bantu yang mampu bekerja baik dengan pustaka lainnya. Setiap fitur dapat dimodifikasi sesuai dengan kebutuhan aplikasi. Menjadi salah satu kerangka kerja yang memfasilitasi pembangunan aplikasi kompleks yang terorganisir dan mudah dirawat. Angular JS lebih dikenal pada kemampuannya melayani sebuah situs web yang menggunakan halaman tunggal untuk penyajian data yang beragam. [4][5]

```
<!DOCTYPE html>
<html lang="en-US">
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
<body>

<div ng-app="">
  <p>Name : <input type="text" ng-model="name"></p>
  <h1>Hello {{name}}</h1>
</div>

</body>
</html>
```

Gambar 2.2: Contoh Penggunaan Angular JS pada Halaman Sederhana

Dengan menggunakan berkas JavaScript Angular yang didapatkan dari CDN (*Content Delivery Network*) atau media lain, pengembang dapat dengan mudah menggunakan fitur yang ditawarkan Angular JS.

2.4 MongoDB

Merupakan salah satu NoSQL (Not only SQL) terkenal yang dirancang untuk mengelola polimorfik, obyek, dan struktur data yang terus berkembang. MongoDB adalah basis data open-source yang memungkinan mengubah skema dengan cepat sementara fungsi yang diharapkan dari basis data tradisional masih berjalan. [6] [7]

Gambar 2.3: Contoh Bentuk Penyimpanan Data MongoDB

Model penyimpanan yang menyerupai JSON membuat pengguna dapat dengan mudah mengakses dan mengubah data yang ada. Karena penyimpanannya yang menyerupai JSON, membuat struktur penyimpanan dapat berubah sewaktu-waktu tanpa mengubah konfigurasi sebelumnya.

2.5 Apache JMeter

Menjadi salah satu alat bantu untuk melakukan tes muat dan mengukur performa aplikasi, salah satunya berbasis web. Mampu melakukan pengujian pada berbagai protokol diantaranya Web, FTP, basis data, Mail (SMTP, POP3, IMAP), serta MongoDB. [8]

Gambar 2.4: Contoh Bentuk Penyimpanan Data MongoDB

Apache JMeter berbasis Java. Cara kerjanya yang menyerupai sebuah browser, karena desain awal memang ditujukan untuk menguji aplikasi berbasis web, mampu mengakses halaman website yang memiliki sumber daya statis maupun dinamis. Namun ada beberapa fitur browser yang tidak ditiru oleh Apache JMeter.

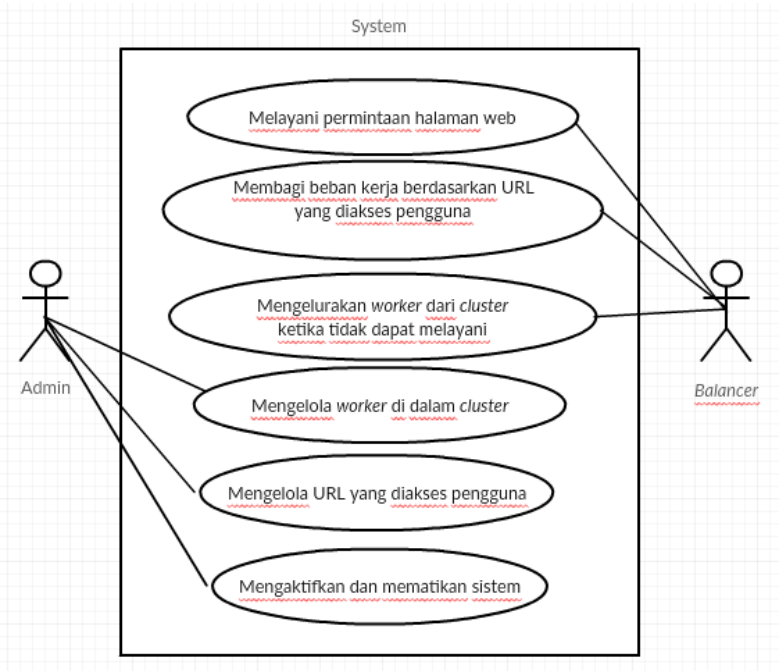
BAB 3

DESAIN DAN PERANCANGAN

Pada bab ini dibahas mengenai analisis dan perancangan sistem.

3.1 Kasus Penggunaan

Terdapat dua aktor dengan masing-masing tiga aktivitas dalam sistem yang digambarkan pada Gambar 3.2.



Gambar 3.1: Digram Kasus Penggunaan

Digram kasus penggunaan pada Gambar 3.2 dideskripsikan masing-masing pada Tabel 3.4.

Tabel 3.1: Daftar Kode Kasus Penggunaan

Kode Kasus Peng-guna-an	**Nama Kasus Penggunaan**	**Keterangan**
UC-0001	Melayani permintaan halaman web	Setiap permintaan halaman web akan mengarah ke sistem dan *balancer* akan melayani dengan cara meneruskan permintaan ke *worker* dan mengeruskan balasan ke pengguna
UC-0002	Membagi beban kerja berdasarkan URL yang diakses pengguna	Dibelakang sistem terdapat beberapa *worker* yang bekerja melayani permintaan terusan dari sistem, di sini lah *balancer* membagi beban kerja tersebut
UC-0003	Mengeluarkan *worker* dari *cluster* ketika tidak dapat melayani	Jika terdapat *worker* yang tidak dapat melayani permintaan, *balancer* akan mengeluarkan sementara dari tugas melayani hingga *worker* mampu memberikan layanan
UC-0004	Mengelola *worker* di dalam *cluster*	Admin dapat menambahkan dan mengurangi *worker* di dalam daftar *cluster*
UC-0005	Mengelola URL yang diakses pengguna	Admin dapat menambahkan dan mengurangi daftar URL ke dalam sistem

Tabel 3.1: Daftar Kode Kasus Penggunaan

Kode Kasus Peng-guna-an	**Nama Kasus Penggunaan**	**Keterangan**
UC-0006	Mengaktifkan dan mematikan sistem	Admin memiliki kendali atas aktif dan tidaknya sistem

3.2 Arsitektur Sistem

Pada sub-bab ini, dibahas mengenai tahap analisis dan kebutuhan bisnis dan desain dari sistem yang akan dibangun.

3.2.1 Desain Umum Sistem

Sistem load balancing yang akan dibangun menggunakan teknologi Node JS dengan JavaScript sebagai bahasa pemrograman. Sistem ini akan berjalan pada port 80 dan menjadi target utama ketika pengguna akan menggunakan aplikasi berbasis web yang dikembangkan dengan menggunakan load balancer ini. Akan terdapat beberapa komputer pembantu, selanjutnya disebut *worker*, yang menerima perintah untuk melayani permintaan halaman website dari load balancer. *Worker* akan digolongkan menjadi dua cluster untuk melayani permintaan sesuai dengan URL yang diakses pengguna. Di sinilah algoritma berbasis konten diterapkan.

Load balancer memiliki daftar worker dan URL yang akan digunakan dalam operasional load balancing. Untuk memudahkan administrator sistem dalam mengkonfigurasikan worker dan menambahkan daftar URL yang ada, disediakan sebuah halaman admin yang berjalan pada port 3000. Selain dengan halaman admin ini, administrator sistem sudah dibantu untuk mengeluarkan worker yang

tidak aktif dari pekerjaannya dalam proses load balancing oleh sistem secara otomatis dalam rentang waktu tertentu.

3.2.2 Desain Rinci Balancer

3.3 Deskripsi Umum Kebutuhan

Tugas Akhir ini membangun sebuah sistem load balancing dengan halaman web sebagai media untuk mengatur cluster yang bekerja dan URL yang digunakan di dalam aplikasi. Diperlukan rancangan-rancangan kebutuhan fungsional dan non-fungsional, yang akan dijelaskan dalam bentuk diagram kasus penggunaan, diagram aktivitas, dan penjelasan secara rinci tentang kebutuhan fungsional dan non-fungsional.

3.3.1 Deskripsi Kebutuhan Fungsional

Kebutuhan fungsional merupakan gambaran dari informasi yang terjadi pada sistem yang akan dibangun. Sistem ini digunakan untuk membagi beban kerja pada aplikasi berbasis web. Dan terdapat sebuah halaman admin untuk memudahkan administrator sistem dalam mengaturkan sumber daya yang ada.

Tabel 3.2: Kebutuhan Fungsional

Kode Kebutuhan Fungsional	**Kebutuhan Fungsional**
F-0001	Melayani permintaan halaman web
F-0002	Membagi beban kerja berdasarkan URL yang diakses pengguna
F-0003	Mengeluarkan *worker* dari *cluster* ketika tidak dapat melayani
F-0004	Mengelola *worker* di dalam *cluster*
F-0005	Mengelola URL yang diakses pengguna
F-0006	Mengaktifkan dan mematikan sistem

3.3.2 Deskripsi Kebutuhan Non-Fungsional

Adapun kebutuhan non-fungsional mendeskripsikan atribut kualitas dari sistem yang dibangun.

3.3.3 Karakteristik Pengguna

Sistem yang dibangun dalam Tugas Akhir ini hanya memiliki satu aktor utama yaitu administrator sistem, selanjutnya disebut admin. Dalam kenyataannya admin adalah seorang manusia. Aktor lain di luar manusia adalah *load balancer*, selanjutnya disebut *balancer*. Penjelasan tentang aktor dalam sistem dijabarkan pada Tabel 3.3.

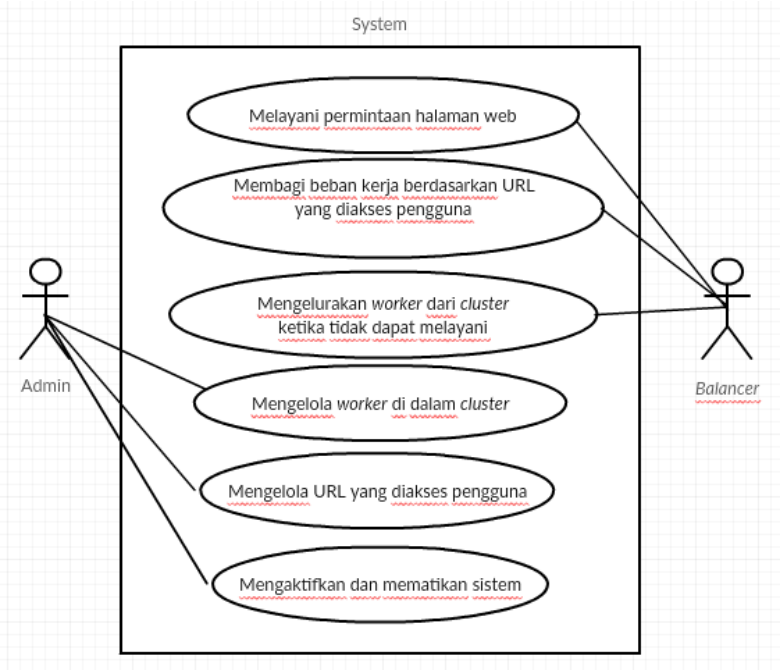
Tabel 3.3: Karakteristik Pengguna

Nama Aktor	**Tugas**	**Hak Akses Aplikasi**
Admin	Mengelola sistem *load balancer*	Mematikan dan mengaktifkan sistem, mengelola *cluster*, mengelola URL
Balancer	Menyediakan layanan web	Menyediakan layanan web dan membagi beban kerja

3.3.4 Digram Kasus Penggunaan

Kebutuhan fungsional yang telah dirancang dalam pembahasan sebelumnya dimodelkan dalam bentuk diagram kasus penggunaa. Terdapat dua aktor dengan masing-masing tiga aktivitas dalam sistem yang digambarkan pada Gambar 3.2.

Digram kasus penggunaan pada Gambar 3.2 dideskripsikan masing-masing pada Tabel 3.4.



Gambar 3.2: Digram Kasus Penggunaan

Tabel 3.4: Daftar Kode Kasus Penggunaan

Kode Kasus Peng-guna-an	**Nama Kasus Penggunaan**	**Keterangan**

Tabel 3.4: Daftar Kode Kasus Penggunaan

Kode Kasus Peng-guna-an	**Nama Kasus Penggunaan**	**Keterangan**
UC-0001	Melayani permintaan halaman web	Setiap permintaan halaman web akan mengarah ke sistem dan *balancer* akan melayani dengan cara meneruskan permintaan ke *worker* dan mengeruskan balasan ke pengguna
UC-0002	Membagi beban kerja berdasarkan URL yang diakses pengguna	Dibelakang sistem terdapat beberapa *worker* yang bekerja melayani permintaan terusan dari sistem, di sini lah *balancer* membagi beban kerja tersebut
UC-0003	Mengeluarkan *worker* dari *cluster* ketika tidak dapat melayani	Jika terdapat *worker* yang tidak dapat melayani permintaan, *balancer* akan mengeluarkan sementara dari tugas melayani hingga *worker* mampu memberikan layanan

Tabel 3.4: Daftar Kode Kasus Penggunaan

Kode Kasus Penggunaan	**Nama Kasus Penggunaan**	**Keterangan**
UC-0004	Mengelola *worker* di dalam *cluster*	Admin dapat menambahkan dan mengurangi *worker* di dalam daftar *cluster*
UC-0005	Mengelola URL yang diakses pengguna	Admin dapat menambahkan dan mengurangi daftar URL ke dalam sistem
UC-0006	Mengaktifkan dan mematikan sistem	Admin memiliki kendali atas aktif dan tidaknya sistem

3.3.5 Windows

Cara mudah untuk memasang distribusi LaTeXdi Windows adalah dengan menggunakan paket MikTeX. Anda dapat mengunduh kakas pemasang (*installer*) MikTeX melalui pranala `http://miktex.org/download`. Pemasang berukuran sekitar 200 MB dan terdiri dari beberapa paket dasar saja (dengan sebuah kakas editor bantu bernama TeXworks). Jika dokumen ingin dikompilasi menggunakan paket yang belum terpasang, MikTeXakan secara otomatis menguduhnya dari Internet sehingga Anda tidak perlu khawatir untuk memasang paket secara manual.

Selain pemasang pada pranala di atas, MikTeXjuga menyediakan paket lengkap berupa DVD yang berisi semua paket LaTeX yang terdaftar di CTAN. Namun sayangnya, DVD tersebut tidak tersedia melalui pengunduhan secara bebas. Anda dapat menghubungi penulis jika berminat mendapatkan DVD ini.

3.3.6 Linux

Sistem operasi Linux umumnya menyediakan cara yang mudah untuk memasang paket TEX Live. Jika Anda menggunakan Ubuntu, Anda dapat memasang paket ini secara penuh melalui perintah `sudo apt-get install texlive-full`. Jika Anda hanya memasang paket dasar saja, Anda dapat memasang paket `texlive` saja (tanpa ada embel-embel apapun).

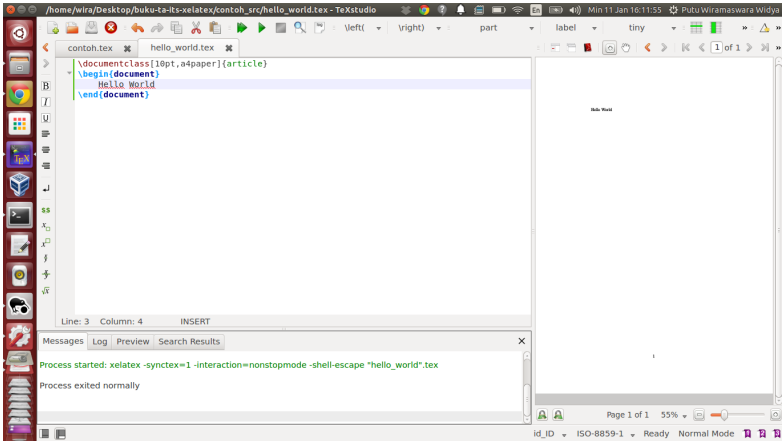
3.3.7 Mac OS X

Pengguna Mac OS dapat menggunakan paket MacTEXyang tersedia melalui pranala `https://tug.org/mactex/`. Paket berukuran 2,4GB ini sudah lebih dari cukup untuk penulisan dokumen LaTEX dasar.

3.4 `Hello World` menggunakan LATEX

Pembuatan dokumen LaTEXmungkin sangat rumit bagi pemula karena membutuhkan penggunaan antarmuka teks (Command Line Interface) pada sistem operasi untuk memanggil *compiler*. Jika Anda tidak ingin bersusah payah dalam hal ini, Anda dapat langsung menggunakan editor yang memang sudah terdedikasi untuk pembuatan dokumen TEX. Editor yang saya sarankan dalam hal ini adalah TEXstudio yang tersedia untuk tiga sistem operasi (Unduh melalui `http://texstudio.sourceforge.net/`). Tangkapan layar dari aplikasi ini dapat dilihat pada Gambar 3.3.

Jika sudah siap, silahkan membuka teks editor favorit Anda dan mulai menulis beberapa bagian teks seperti pada Gambar 3.4. Anda boleh menggunakan atau tidak indentasi pada setiap elemen. Penggunaan indentasi dalam hal ini bermaksud untuk memudahkan pembacaan struktur dokumen. Simpan berkas tersebut ke dalam berkas bernama ”hello_world.tex”.



Gambar 3.3: Tangkapan Layar TEXstudio

```
\documentclass[10pt,a4paper]{article}
\begin{document}
    Hello World
\end{document}
```

Gambar 3.4: Artikel Hello World

3.4.1 Kompilasi Dokumen

Untuk memroses kode ke dokumen, Anda dapat menggunakan menu `Tools | Build and View (F1)` pada TEXstudio atau memanggil perintah berikut pada antarmuka teks (jika Anda sudah terbiasa dan pastikan berada pada direktori yang tepat) :

```
latex hello_world.tex
```

Setelah beberapa pesan kompilasi muncul, Anda dapat membuka berkas ”hello_world.dvi” yang merupakan dokumen hasil kompilasi.

Jika Anda ingin membuat dokumen dalam format PDF, Anda

dapat menggunakan kompilator bernama `pdflatex`. Hasilnya dapat dilihat pada Gambar 3.5 Kompilator bawaan dapat Anda ubah dalam TEXstudio melalui menu `Options | Configure TeX Studio | Build | Default Compiler`. Untuk lebih jelasnya, distribusi TEXatau LATEXumumnya memiliki kompilator sebagai berikut :

- **LATEX** merupakan kompilator bawaan untuk LATEXyang merupakan pengembangan dari TEX (dikembangkan oleh Donald Knuth). Fitur utama LATEX antara lain: pilihan kelas dokumen dan adanya strukturisasi dokumen.
- **pdfLATEX** merupakan pengembangan dari LATEXyang akan menghasilkan luaran dalam bentuk PDF ketimbang DVI.
- **XeLATEX** merupakan pengembangan dari LATEXdengan dukungan fonta berbasis TrueType. Templat Buku TA ini menggunakan XeLATEXagar dapat menggunakan fonta **Times New Roman** bawaan dari Windows.

Gambar 3.5: hello_world.pdf

3.4.2 Struktur Kode LaTeX

Secara umum, struktur kode berkas .tex dibagi menjadi dua bagian:

- **Preambule**, merupakan bagian yang berada sebelum `\begin{document}` dilakukan. Pada bagian ini, biasanya diawali dengan deklarasi `\documentclass` (dibahas selanjutnya) dan deklarasi impor paket tambahan yang dibutuhkan (melalui `\usepackage{}`).
- **Dokumen** merupakan bagian yang diawali dengan `\begin{document}` dan diakhiri dengan `\end{document}`. Pada bagian inilah Anda mengisi konten dari dokumen Anda secara berurutan per halamannya.

3.4.3 Kelas Dokumen Bawaan

Anda tidak seharusnya memikirkan bagaimana templat dan tata letak dokumen Anda di LaTeX jika Anda memang fokus untuk menulis dokumen. Filosofi di LaTeX menegaskan bahwa Anda memang harus fokus terhadap isi konten daripada terdistraksi dengan bagaimana wujud dokumen ketika dicetak. Untuk tujuan ini, LaTeX beserta para komunitas menyediakan banyak templat untuk banyak keperluan yang bisa digunakan oleh pengguna sehingga mereka bisa langsung fokus mengisi konten dari dokumen mereka. Jenis templat ini dapat Anda pilih pada bagian `\documentclass{`**`namatemplat`**`}`.

LaTeX sendiri memiliki beberapa templat bawaan :

- **article**: templat untuk artikel ilmiah.
- **book**: templat untuk penulisan buku, terdapat struktur dokumen layaknya buku seperti `part` dan `chapter`.
- **report**: seperti templat **book** tapi tidak memiliki pembagian halaman awal, halaman isi dan lampiran.
- **letter**: untuk penulisan dokumen tanpa struktur di dalamnya.

Anda dapat menambahkan beberapa opsi pada templat yang Anda pilih dengan menambahkan argumen opsional (menggunakan ku-

rung siku) pada deklarasi `\documentclass`. Misalnya, jika Anda ingin membuat dokumen buku dengan ukuran kertas A5 dengan ukuran fonta 11pt (seperti Buku TA), Anda bisa membuat deklarasi sebagai berikut:

```
\documentclass[a5paper,11pt]{book}
```

Selain templat bawaan LaTeX, terdapat beberapa templat yang disediakan oleh komunitas:

- **IEEEtran** merupakan templat untuk menulis jurnal dalam format IEEE. Bisa digunakan juga untuk pengiriman jurnal ilmiah POMITS. (Contoh pada Gambar 3.6)
- **beamer** merupakan templat untuk membuat *slide* presentasi. (Contoh pada Gambar 3.7)
- **a0poster** merupakan templat untuk membuat dokumen dengan ukuran kertas yang besar, seperti poster.
- **memoir** merupakan sekumpulan templat untuk kebutuhan penulisan buku fiksi maupun non-fiksi.
- **moderncv** untuk penulisan Curriculum Vitae.
- dan masih banyak lagi.

3.5 Cara Menggunakan Templat `ta-its`

Berkas `ta-its.cls` yang disertakan pada templat ini merupakan bagian utama dari templat Buku TA ITS yang siap untuk digunakan. Untuk menggunakan templat ini, salinlah berkas `ta-its.cls` dan direktori `img/` (berisi berkas sampul) ke direktori di mana Anda akan menulis Buku TA Anda. Kemudian gunakan templat ini sebagai kelas dokumen melalui deklarasi sebagai berikut pada Preambule :

```
\documentclass{buku-ta}
```

Templat ini tidak menerima argumen tambahan apapun untuk saat ini. Selanjutnya, Anda wajib mendeklarasikan Judul, Pengarang, Dosen dan Jurusan juga pada bagian Preambule. Formatnya adalah sebagai berikut :

- `\title{Judul TA dalam Bahasa Indonesia}{Judul`

JURNAL TEKNIK POMITS VOL. 2, NO. 1, (2013) ISSN: 2337-3539 (2301-9271 PRINT) 1

Rancang Bangun Layanan Platform as a Service (PAAS) untuk Mendukung Sistem Multi-Tenancy Pengembangan Aplikasi Berbasis Komputasi Awan

Putu Wiramaswara Widya*, Royyana Muslim Ijtihadie† dan Baskoro Adi Pratomo‡
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
Email: *wiramaswara11@mhs.if.its.ac.id, †roy@if.its.ac.id, ‡baskoro@if.its.ac.id

***Ringkasan*—Tugas Akhir ini merancang sebuah sistem Platform as a Service (PaaS) untuk kebutuhan Web Hosting atau penempatan aplikasi di Internet yang bersifat *multi-tenancy* (multi pengguna) dengan dukungan empat *platform* pengembangan aplikasi Web yaitu PHP 5.5, Node.js, Python 2.7 dan Ruby 1.9; dukungan pembagian muat aplikasi melalui aplikasi penyeimbang muat (*load balancer*) dan mengadopsi tiga prinsip komputasi awan, yaitu: *self-service*, *resource pooling* dan *measured service*. Sistem ini dikembangkan untuk membuka peluang bagi jasa layanan Web Hosting untuk menyediakan layanan Hosting yang lebih kompetitif dengan fitur yang sesuai dengan perkembangan Komputasi Awan saat ini.**

Sistem dikembangkan menggunakan platform pengembangan aplikasi MEAN Framework (MongoDB, Express.js, Angular.js

Selain Shared Web Hosting dan VPS, saat ini terdapat jenis web hosting yang mulai populer yaitu Cloud Web Hosting. Layanan ini mengadopsi prinsip komputasi awan yaitu berupa Platform as a Service (PaaS) yang menyediakan jasa platform bagi pengembang Web untuk menempatkan aplikasi mereka di Internet. Layanan ini memberikan fitur tambahan yang menjadi karakteristik komputasi awan: transparansi akses, *multi-platform*, skalabilitas, reliabilitas, keterbukaan akses API dan kemudahan penggunaan. Contoh layanan semacam ini adalah Microsoft Azure, Heroku, OpenShift, Google Apps Engine. Sayangnya, layanan ini masih sangat jarang ada terutama di Indonesia. Kebanyakan layanan yang dipasarkan sebagai

Gambar 3.6: Contoh penggunaan templat IEEETran

```
TA dalam Bahasa Inggris}
```

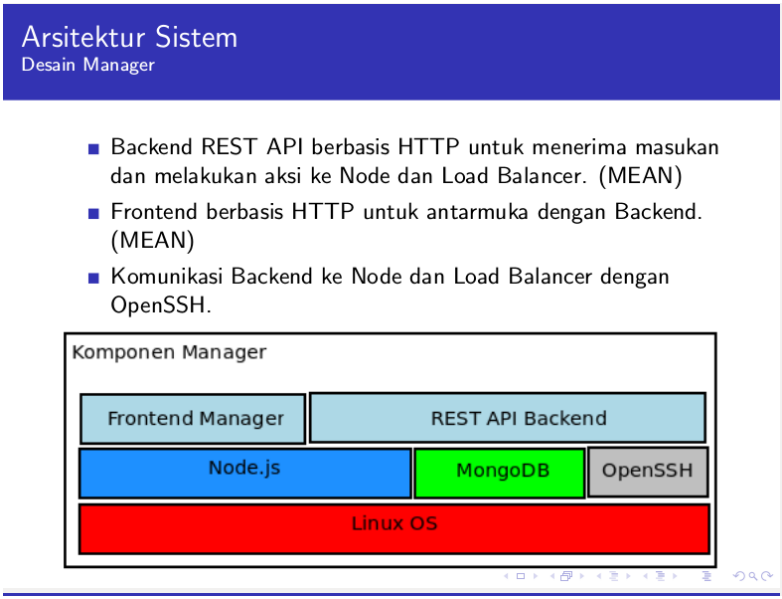
- `\author{Nama Penulis}{NRP Penulis}`
- `\degree{Nama Gelar}{Bidang Studi}{Program Studi}{Jurusan}{Jurusan (English)}{Fakultas}{Fakultas Singkatan}{Fakultas (English)}`
- `\time{Bulan Pembuatan}{Tahun Pembuatan}`

Kemudian pada bagian isi, templat ini menawarkan beberapa fungsi untuk pembuatan elemen buku secara otomatis, antara lain :

- `\maketitle` digunakan untuk membuat sampul dalam tiga halaman: Sampul Depan, Sampul Tengah, Sampul Tengah Bahasa Inggris.
- `\legalityPaper` untuk membuat halaman pengesahan.
- Environment `abstrak` dan `abstract` untuk penulisan Abstrak dalam Bahasa Indonesia dan Inggris.

Contoh penggunaan dapat Anda lihat pada berkas `contoh.tex`.

Dengan menggunakan templat ini, Anda akan menghemat waktu Anda untuk membuat sampul dan halaman pengesahan yang ka-



Gambar 3.7: Contoh penggunaan templat beamer

dang bisa membuat kerepotan dalam hal pengaturan posisi indentasinya.

3.6 Struktur Dokumen LaTeX

Dokumen LaTeX terdiri dari struktur yang dibuat berdasarkan struktur dokumen sehari-hari. Sebagai penulis dokumen, Anda wajib menggunakan struktur ini sehingga LaTeX dapat melakukan hal lain yang membantu Anda dalam mengorganisir dokumen seperti misalnya pembuatan Daftar Isi. Berikut adalah struktur dokumen yang ada di LaTeX diurutkan berdasarkan hirarkinya.

Tabel 3.5: Struktur hirarki dokumen LaTeX

Nama	**Peruntukkan**
\part{Judul Bagian}	`book`
\chapter{Judul Bab}	`book` dan `report`
\section{Judul Subbab}	semua kecuali `letter`
\subsection{Judul Subsubbab}	semua kecuali `letter`
\subsubsection{Judul Subsubsubbab}	semua kecuali `letter`
\paragraph{Judul Paragraf}	semua

Untuk templat pihak ketiga, Anda dapat melihat dokumentasi dari templat bersangkutan. Sebagai informasi, templat `ta-its` dibuat berdasarkan templat `book` sehingga struktur dokumennya sama.

3.7 Paragraph dan Teks

3.8 Daftar

3.9 Gambar

3.10 Tabel

3.11 Rumus Matematika

3.12 Algoritma

3.13 Kode Sumber