



**UNIVERSITAS INDONESIA**

**JUDUL SESUATU BANGET *ENGLISH* MIRING JUGA**

**SKRIPSI**

**ANDRI KURNIAWAN**

**1306382064**

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER  
DEPOK  
JUNI 2017**



**UNIVERSITAS INDONESIA**

**JUDUL SESUATU BANGET *ENGLISH* MIRING JUGA**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Ilmu Komputer**

**ANDRI KURNIAWAN**

**1306382064**

**FAKULTAS ILMU KOMPUTER  
PROGRAM STUDI ILMU KOMPUTER  
DEPOK  
JUNI 2017**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Andri Kurniawan**  
**NPM : 1306382064**  
**Tanda Tangan :**

**Tanggal : 5 Juni 2017**

## **HALAMAN PENGESAHAN**

Skripsi ini diajukan oleh :

Nama : Andri Kurniawan  
NPM : 1306382064  
Program Studi : Ilmu Komputer  
Judul Skripsi : Pembentukan Web

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Ilmu Komputer pada Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Indonesia.**

## **DEWAN PENGUJI**

Pembimbing : Dr. Ade Azurat ( )

Penguji : Penguji 1 ( )

Penguji : Penguji 2 ( )

Ditetapkan di : Depok

Tanggal : 5 Juli 2017

## KATA PENGANTAR

*Alhamdulillahirabbil'alamin*, segala puji dan syukur kehadiran Tuhan Yang Maha Esa, Allah Subhana Huwataala, karena hanya dengan hidayah dan rahmat-Nya, penulis dapat menyelesaikan pembuatan skripsi ini.

*Allahumma sholli 'alaa sayyidina Muhammad*, Sholawat serta salam tak henti-hentinya dipanjatkan kepada Rasulullah SAW, atas peranannya di muka bumi dalam memberikan tuntunan kepada seluruh umat manusia, dan sebagai inspirasi atas seluruh manusia sebagai manusia dengan akhlak terbaik.

Penulisan skripsi ini ditujukan untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan pada Program Sarjana Ilmu Komputer, Universitas Indonesia. Saya sadar bahwa dalam perjalanan menempuh kegiatan penerimaan dan adaptasi, belajar-mengajar, hingga penulisan skripsi ini, penulis tidak sendirian. Penulis ingin berterima kasih kepada pihak-pihak berikut :

Depok, 17 Juni 2017

Andri Kurniawan

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

**Nama** : Andri Kurniawan  
**NPM** : 1306382064  
**Program Studi** : Ilmu Komputer  
**Fakultas** : Ilmu Komputer  
**Jenis Karya** : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

Pembentukan Web

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 5 Juni 2017  
Yang menyatakan

(Andri Kurniawan)

## **ABSTRAK**

Nama : Andri Kurniawan  
Program Studi : Ilmu Komputer  
Judul : Pembentukan Web

Abstrak INA

Kata Kunci:  
atu, dua, *tiga*

## **ABSTRACT**

Name : Andri Kurniawan  
Program : Computer Science  
Title : Sesuatu Banget in English

Abstract in Eng

Keywords:  
one,two,three



## DAFTAR ISI

<b>HALAMAN JUDUL</b>	<b>i</b>
<b>LEMBAR PERNYATAAN ORISINALITAS</b>	<b>ii</b>
<b>LEMBAR PENGESAHAN</b>	<b>iii</b>
<b>KATA PENGANTAR</b>	<b>iv</b>
<b>LEMBAR PERSETUJUAN PUBLIKASI ILMIAH</b>	<b>v</b>
<b>ABSTRAK</b>	<b>vi</b>
<b>Daftar Isi</b>	<b>viii</b>
<b>Daftar Gambar</b>	<b>x</b>
<b>Daftar Tabel</b>	<b>xi</b>
<b>Daftar Kode</b>	<b>xii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Perumusan Masalah . . . . .	1
1.3 Tujuan dan Manfaat Penelitian . . . . .	1
1.4 Tahapan Penelitian . . . . .	1
1.5 Ruang Lingkup Penelitian . . . . .	2
1.6 Sistematika Penulisan . . . . .	2
<b>2 TINJAUAN PUSTAKA</b>	<b>3</b>
2.1 X si sesuatu . . . . .	3
2.1.1 Pengertian X . . . . .	3
2.1.2 Klasifikasi X . . . . .	4
2.2 <i>Section in Eng</i> . . . . .	5
2.2.1 Pengertian <i>Section in Eng</i> . . . . .	5
2.2.2 Next Subsection <i>Section in Eng</i> . . . . .	5
2.3 <i>Keatas lagi</i> . . . . .	5
2.3.1 <i>Masuk lagi</i> . . . . .	6
<b>3 DESAIN</b>	<b>7</b>
3.1 Desain Sistem . . . . .	7
3.2 Desain <i>Bussiness Logic</i> . . . . .	8
3.3 Desain <i>Rules</i> . . . . .	8

<b>4</b>	<b>IMPLEMENTASI</b>	<b>10</b>
4.1	Instalasi Zotonic . . . . .	10
4.2	Pembuatan Situs baru dengan Zotonic . . . . .	11
4.3	Implementasi Pemanggilan <i>Web Service</i> . . . . .	11
4.4	Implementasi <i>Rules</i> . . . . .	11
<b>5</b>	<b>HASIL</b>	<b>12</b>
5.1	Implementasi <i>Cluster</i> . . . . .	12
5.1.1	Instalasi <i>Frontend</i> . . . . .	12
5.1.2	Konfigurasi . . . . .	14
5.1.2.1	semakin ke dalam . . . . .	15
5.2	Pengujian . . . . .	15
5.2.1	Kasus Uji . . . . .	15
5.2.2	Kasus Uji . . . . .	16
<b>6</b>	<b>PENUTUP</b>	<b>17</b>
6.1	Kesimpulan . . . . .	17
6.2	Saran . . . . .	17
	<b>LAMPIRAN</b>	<b>1</b>
	<b>Lampiran 1 : Kode Sumber</b>	<b>2</b>
	<b>Lampiran 2 : Berkas Konfigurasi</b>	<b>2</b>
	<b>Lampiran 8 : UAT dan Kuesioner</b>	<b>3</b>

## DAFTAR GAMBAR

2.1	Contoh masalah yang dikerjakan secara paralel . . . . .	4
2.2	Arsitektur klasik <i>von Neumann</i> . . . . .	4
3.1	Desain sistem secara keseluruhan . . . . .	7

## DAFTAR TABEL

2.1	Fungsi fundamental MPI . . . . .	6
4.1	Contoh Tabel . . . . .	10
4.2	An Example of Rows Spanning Multiple Columns . . . . .	10
4.3	An Example of Columns Spanning Multiple Rows . . . . .	10
4.4	An Example of Spanning in Both Directions Simultaneously . . . . .	11
5.1	Informasi <i>cluster</i> X . . . . .	12
5.2	Perbandingan Partisi <i>default</i> dan manual . . . . .	13
1	Tabel UAT dan Kuesioner . . . . .	4

## DAFTAR KODE

5.1	Keluaran output . . . . .	13
5.2	Keluaran mentah untuk detail <i>job</i> . . . . .	15
5.3	Potongan skrip submisi <i>job</i> melalui torqace . . . . .	15
5.4	Potongan Makefile <i>project</i> . . . . .	16
1	Skrip menambahkan pengguna baru . . . . .	2
2	<i>Cronjob</i> menambahkan pengguna baru . . . . .	2
3	Berkas <code>compute.xml</code> . . . . .	3

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Menurut ? terdapat 3 buah contoh untuk membuat enumerate pada latex (?):

1. Makan
2. Minum

Menurut ?, pemodelan yang sama apabila dijalankan dengan komputer *Dual Core* maka akan membutuhkan waktu 1 tahun dengan asumsi memori yang dibutuhkan cukup (?).

### 1.2 Perumusan Masalah

Pada bagian ini akan dijelaskan mengenai definisi permasalahan yang dihadapi dan ingin diselesaikan serta asumsi dan batasan yang digunakan dalam menyelesaikannya.

### 1.3 Tujuan dan Manfaat Penelitian

Dibawah ini adalah contoh itemize :

- Terimplementasinya .
- Menyelesaikan masalah .

### 1.4 Tahapan Penelitian

<b>@todo</b>
Tuliskan tujuan penelitian.

## 1.5 Ruang Lingkup Penelitian

## 1.6 Sistematika Penulisan

Sistematika penulisan laporan adalah sebagai berikut:

- Bab 1 PENDAHULUAN
- Bab 2 TINJAUAN PUSTAKA
- Bab 3 DESAIN
- Bab 4 IMPLEMENTASI
- Bab 5 HASIL
- Bab 6 PENUTUP

**@todo**

Tambahkan penjelasan singkat mengenai isi masing-masing bab.

## BAB 2

### TINJAUAN PUSTAKA

Bab ini berisi .... Pada sub-bab 2.1 akan dijelaskan dasar-dasar ...

#### 2.1 X si sesuatu

Dokumen  $\text{\LaTeX}$  sangat mudah, seperti halnya membuat dokumen teks biasa. Ada beberapa perintah yang diawali dengan tanda `'\'`. Seperti perintah `\\` yang digunakan untuk memberi baris baru. Perintah tersebut juga sama dengan perintah `\newline`. Pada bagian ini akan sedikit dijelaskan cara manipulasi teks dan perintah-perintah  $\text{\LaTeX}$  yang mungkin akan sering digunakan. Jika ingin belajar hal-hal dasar mengenai  $\text{\LaTeX}$ , silahkan kunjungi:

- <http://frodo.elon.edu/tutorial/tutorial/>, atau
- <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/>

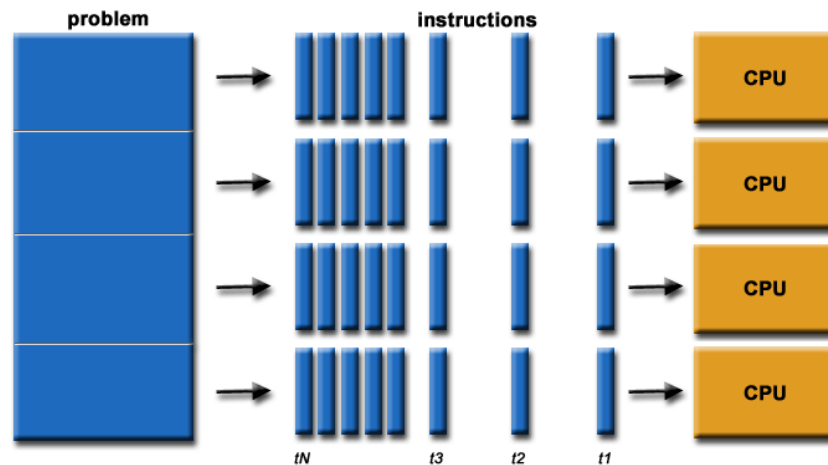
##### 2.1.1 Pengertian X

Setiap gambar dapat diberikan caption dan diberikan label. Label dapat digunakan untuk menunjuk gambar tertentu. Jika posisi gambar berubah, maka nomor gambar juga akan diubah secara otomatis. Begitu juga dengan seluruh referensi yang menunjuk pada gambar tersebut.

Contoh sederhana adalah Gambar 2.1. Silahkan lihat code  $\text{\LaTeX}$  dengan nama `bab2-landasan-teori.tex` untuk melihat kode lengkapnya. Harap diingat bahwa caption untuk gambar selalu terletak dibawah gambar.

Dibawah `adda figure`, jangan lupa dimension dengan 2.1.





**Gambar 2.1:** Contoh masalah yang dikerjakan secara paralel

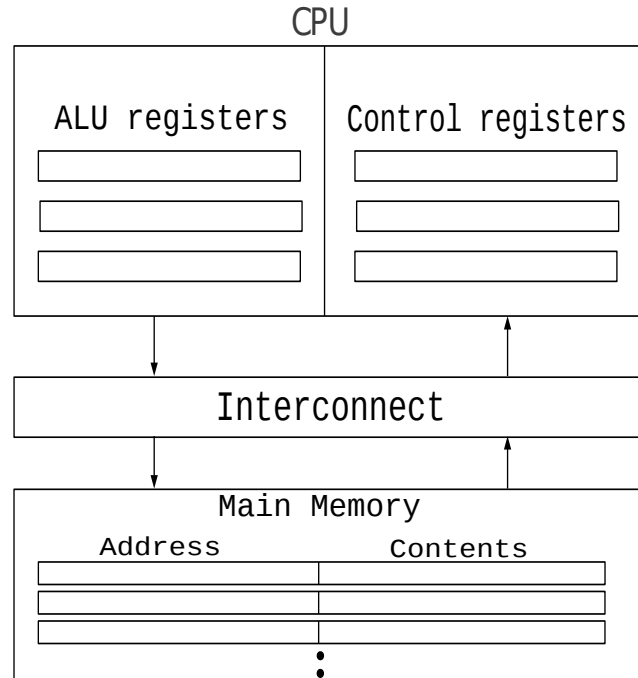
Sumber gambar: (?)

### 2.1.2 Klasifikasi X

Figure dalam enum dan dua sitasi sekaligus (??) :

#### 1. ***Bold Italic***

Penjelasan..... Untuk gambarannya dapat dilihat di Gambar 2.2.



**Gambar 2.2:** Arsitektur klasik von Neumann

Sumber gambar terinspirasi dari: (?)

#### 2. ***Sesuatu banget***

Penjelasan.....

## 2.2 *Section in Eng*

Hal pertama yang mungkin ditanyakan adalah bagaimana membuat huruf tercetak tebal, miring, atau memiliki garis bawah. Pada Texmaker, Anda bisa melakukan hal ini seperti halnya saat mengubah dokumen dengan LO Writer. Namun jika tetap masih tertarik dengan cara lain, ini dia:

- **Bold**  
Gunakan perintah `\textbf{}` atau `\bo{}`.
- *Italic*  
Gunakan perintah `\textit{}` atau `\f{}`.
- Underline  
Gunakan perintah `\underline{}`.
- Overline  
Gunakan perintah `\overline{}`.
- *superscript*  
Gunakan perintah `\{}`.
- *subscript*  
Gunakan perintah `\_{}`.

Perintah `\f` dan `\bo` hanya dapat digunakan jika package `uithesis` digunakan.

### 2.2.1 *Pengertian Section in Eng*

### 2.2.2 *Next Subsection Section in Eng*

## 2.3 *Keatas lagi*

Contoh cite yang ga ada ?. Cite author ?,cite tahun ?, cite mention ?, dan cite di akhir kalimat (?).

### 2.3.1 Masuk lagi

Footnote example nih : MPICH <sup>1</sup>, LAM/MPI <sup>2</sup>, dan OpenMPI <sup>3</sup> (?). MPI-3 sedang dalam tahap perencanaan <sup>4</sup>. Fungsi-fungsi tersebut berada di tabel 2.1. (Contoh tabel).

**Tabel 2.1:** Fungsi fundamental MPI

No.	Nama Fungsi	Penjelasan
1	MPI_Init	Memulai kode MPI
2	MPI_Finalize	Mengakhiri kode MPI
3	MPI_Comm_size	Menentukan jumlah proses
4	MPI_Comm_rank	Menentukan label proses
5	MPI_Send	Mengirim pesan
6	MPI_Recv	Menerima pesan

Sumber tabel: taro sitasi disini, if i were u

---

<sup>1</sup><http://www.mpich.org/>

<sup>2</sup><http://www.lam-mpi.org/>

<sup>3</sup>[www.open-mpi.org](http://www.open-mpi.org)

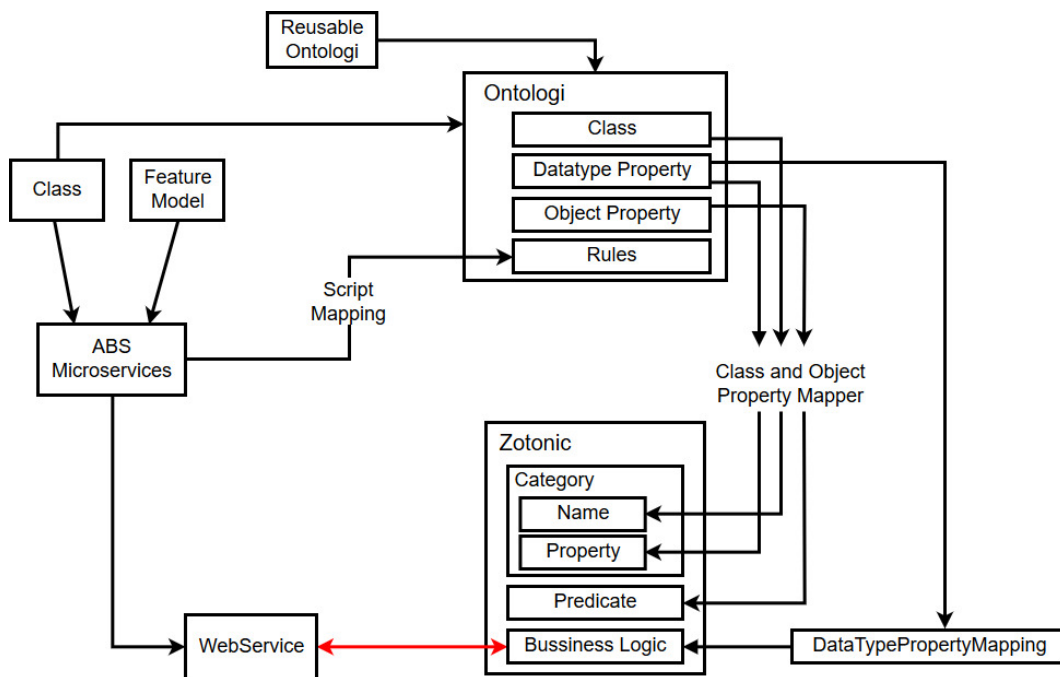
<sup>4</sup>[http://meetings.mpi-forum.org/MPI\\_3.0\\_main\\_page.php](http://meetings.mpi-forum.org/MPI_3.0_main_page.php)

## BAB 3

### DESAIN

#### 3.1 Desain Sistem

Untuk dapat menghasilkan program sesuai tujuan, maka perlu dilakukan desain mengenai sistem secara keseluruhan. Sesuai dengan tujuan dari program yang akan menghubungkan antara *Zotonic* dan *web services* yang dihasilkan oleh *ABS microservices* maka diperlukan sebuah *script* pada *Zotonic* untuk menghubungkan keduanya.



**Gambar 3.1:** Desain sistem secara keseluruhan

Pada Gambar 3.1, terdapat rancangan sistem secara keseluruhan. Pada bagian *Class* dan *Feature Model* untuk menjadi *ABS microservices* telah dilakukan penelitian sehingga penulis tidak perlu melakukan perubahan pada bagian tersebut. Selanjutnya pada bagian *ABS microservices* menjadi *web services* juga telah dilakukan penelitian. Untuk bagian pemetaan dari *Class* menjadi sebuah ontologi dan pemetaan dari *ABS microservices* yang dihasilkan dari *Class* dan *Feature Model* menjadi *Rules* pada ontologi sedang dilakukan penelitian namun hal tersebut tidak menjadi bagian dari penelitian penulis.

Untuk bagian pemetaan dari ontologi ke dalam Zotonic telah dilakukan penelitian. Berdasarkan hasil penelitian tersebut, *Class* pada ontologi akan dipetakan menjadi nama dari *Category* pada Zotonic menggunakan *script classAndObjectPropertyMapper.sh*. *Datatype Property* pada ontologi akan dipetakan menjadi *property* dari *Category* yang didapatkan dari *class* pada ontologi menggunakan *script classAndObjectPropertyMapper.sh*. *Object Property* pada ontologi akan dipetakan menjadi *Predicate* pada Zotonic menggunakan *script classAndObjectPropertyMapper.sh*. Selain dipetakan menjadi *property* dari *Category* pada Zotonic, *Datatype Property* juga digunakan untuk proses pembuatan *bussiness logic* pada Zotonic namun fungsi dari *bussiness logic* tersebut masih dibuat secara manual langsung pada kode program.

### 3.2 Desain *Bussiness Logic*

Beberapa perintah yang dapat digunakan untuk mengubah tampilan adalah:

- `\f`  
Merupakan alias untuk perintah `\textit`, contoh *contoh hasil tulisan*.
- `\bi`  
**Contoh hasil tulisan.**
- `\bo`  
**Contoh hasil tulisan.**
- `\code`  
Contoh hasil tulisan.

### 3.3 Desain *Rules*

Beberapa perintah yang dapat digunakan untuk mengubah tampilan adalah:

- `\f`  
Merupakan alias untuk perintah `\textit`, contoh *contoh hasil tulisan*.
- `\bi`  
**Contoh hasil tulisan.**
- `\bo`  
**Contoh hasil tulisan.**

- `\code`

Contoh hasil tulisan.

## BAB 4

### IMPLEMENTASI

#### 4.1 Instalasi Zotonic

Seperti pada gambar, tabel juga dapat diberi label dan caption. Caption pada tabel terletak pada bagian atas tabel. Contoh tabel sederhana dapat dilihat pada Tabel 4.1.

**Tabel 4.1:** Contoh Tabel

	kol 1	kol 2
baris 1	1	2
baris 2	3	4
baris 3	5	6
jumlah	9	12

Ada jenis tabel lain yang dapat dibuat dengan  $\text{\LaTeX}$  berikut beberapa diantaranya. Contoh-contoh ini bersumber dari <http://en.wikibooks.org/wiki/LaTeX/Tables>

**Tabel 4.2:** An Example of Rows Spanning Multiple Columns

No	Name	Week 1			Week 2		
		A	B	C	A	B	C
1	Lala	1	2	3	4	5	6
2	Lili	1	2	3	4	5	6
3	Lulu	1	2	3	4	5	6

**Tabel 4.3:** An Example of Columns Spanning Multiple Rows

Percobaan	Iterasi	Waktu
Pertama	1	0.1 sec
Kedua	1	0.1 sec
	3	0.15 sec
Ketiga	1	0.09 sec
	2	0.16 sec
	3	0.21 sec

**Tabel 4.4:** An Example of Spanning in Both Directions Simultaneously

		Title			
		A	B	C	D
Type	X	1	2	3	4
	Y	0.5	1.0	1.5	2.0
Resource	I	10	20	30	40
	J	5	10	15	20

## 4.2 Pembuatan Situs baru dengan Zotonic

Berkas ini berisi seluruh berkas Latex yang dibaca, jadi bisa dikatakan sebagai berkas utama. Dari berkas ini kita dapat mengatur bab apa saja yang ingin kita tampilkan dalam dokumen.

## 4.3 Implementasi Pemanggilan *Web Service*

Berkas ini berguna untuk mempermudah pembuatan beberapa template standar. Anda diminta untuk menuliskan judul laporan, nama, npm, dan hal-hal lain yang dibutuhkan untuk pembuatan template.

## 4.4 Implementasi *Rules*

Berkas istilah digunakan untuk mencatat istilah-istilah yang digunakan. Fungsinya hanya untuk memudahkan penulisan. Pada beberapa kasus, ada kata-kata yang harus selalu muncul dengan tercetak miring atau tercetak tebal. Dengan menjadikan kata-kata tersebut sebagai sebuah perintah  $\text{\LaTeX}$  tentu akan mempercepat dan mempermudah pengerjaan laporan.



## BAB 5

### HASIL

#### 5.1 Implementasi *Cluster*

##### 5.1.1 Instalasi *Frontend*

Tabel model lain, ditunjukkan pada tabel 5.1.

**Tabel 5.1:** Informasi *cluster X*

Host Name	X
Cluster Name	X
Certificate Organization	UI
Certificate Locality	Depok
Certificate State	West Java
Certificate Country	ID
Contact	X
URL	<a href="http://grid.ui.ac.id">http://grid.ui.ac.id</a>

Ada pagebreak disini.

Another type of table

**Tabel 5.2:** Perbandingan Partisi *default* dan manual

	Partisi default	Partisi manual yang dilakukan
/	16 GB	30 GB
/var	4 GB	18 GB
swap	1 GB	2 GB
/export	55 GB	26 GB

Program menghasilkan keluaran seperti pada kode 5.1.

**Kode 5.1:** Keluaran output

```
[root@nas-0-0 ~]# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sda4[0] sdb2[1]
      1917672312 blocks super 1.2 [2/2] [UU]

unused devices: <none>
[root@nas-0-0 ~]# mdadm --detail /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Fri May  3 15:38:52 2013
    Raid Level : raid1
    Array Size : 1917672312 (1828.83 GiB 1963.70 GB)
  Used Dev Size : 1917672312 (1828.83 GiB 1963.70 GB)
    Raid Devices : 2
  Total Devices : 2
 Persistence : Superblock is persistent

Update Time : Tue May 28 11:27:49 2013
    State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

    Name : nas-0-0.local:0 (local to host nas-0-0.local)
   UUID : 0754726d:3dfbd4b9:42b0f587:68631556
  Events : 28

   Number   Major   Minor   RaidDevice State
    0         8         4         0     active sync   /dev/sda4
    1         8        18         1     active sync   /dev/sdb2
```

### 5.1.2 Konfigurasi

Contoh verbatim dalam itemize :

- **Bold ini**

dijalankan perintah berikut :

```
# javac Ganteng.java
# java Ganteng
```

#### Perilaku sistem

```
# hai
# enable
# cd /export/rocks/install/
# create distro
# sh sesuatu.sh
# reboot
```

- **Menambahkan *package* pada *compute node***

Langkah yang dilakukan adalah sebagai berikut :

1. Masuk ke dalam direktori `/procfs/`
2. Membuat/Mengubah berkas `xx.xml`. Jika tidak terdapat berkas tersebut, dapat disalin dari `skeleton.xml`.
3. Menambahkan *package* yang ingin dipasang pada *compute node* di-antara *tag* `<package>` seperti berikut : `<package>[package yang akan dipasang]</package>`.
4. Menjalankan perintah berikut termasuk perintah untuk melakukan instalasi ulang seluruh *compute node*:

```
# cd /export/somedir
# create
# run host
```

### 5.1.2.1 semakin ke dalam

**Kode 5.2:** Keluaran mentah untuk detail *job*

```
[ardhi@xx ~]$ qstat -f 138
Job Id: 138.xx
  Job_Name = cur-1000-1np
  Job_Owner = ardhi@xx
  resources_used.cput = 27:21:35
  resources_used.mem = 86060kb
  resources_used.vmem = 170440kb
  resources_used.walltime = 27:24:50
  job_state = R
  queue = default
  server = hastinapura.grid.ui.ac.id
  Checkpoint = u
  ctime = Fri May 31 10:27:37 2013
  Error_Path = xx:/home/ardhi/xx/curcumin-1000/cur-1000-1np.e138
  exec_host = compute-0-5/0
  exec_port = 15003
  Hold_Types = n
  Join_Path = n
  Keep_Files = n
  Mail_Points = e
  Mail_Users = ardhi.putra@ui.ac.id
  mtime = Fri May 31 10:27:47 2013
  Output_Path = xx:/home/ardhi/xx/curcumin-1000/cur-1000-1np.o138
  Priority = 0
  qtime = Fri May 31 10:27:37 2013
  Rerunnable = True
  Resource_List.nodes = 1:ppn=1
  session_id = 5768
  etime = Fri May 31 10:27:37 2013
  submit_args = cur-1000-1np.pbs
  start_time = Fri May 31 10:27:47 2013
  submit_host = xx
  init_work_dir = /home/ardhi/xx/curcumin-1000
```

## 5.2 Pengujian

### 5.2.1 Kasus Uji

Berwarna!

**Kode 5.3:** Potongan skrip submisi *job* melalui torque

```
# Go To working directory
cd $PBS_O_WORKDIR

#openMPI prerequisite
. /opt/torque/etc/openmpi-setup.sh
```

```

mpirun -np 5 -machinefile $PBS_NODEFILE mdrun -v -s \
  curcum400ps.tpr -o md_prod_curcum400_5np.trr -c lox_pr.gro
...

```

### 5.2.2 Kasus Uji

Contoh skrip yang dimasukkan pada *form* yang disediakan dapat dilihat pada kode 5.4.

**Kode 5.4:** Potongan Makefile *project*

```

# Make file for MPI
SHELL=/bin/sh

# Compiler to use
# You may need to change CC to something like CC=mpiCC
# openmpi : mpiCC
# mpich2   : /opt/mpich2/gnu/bin/mpicxx
CC=mpiCC
...
...

```

## **BAB 6**

### **PENUTUP**

Pada bab terakhir ini,

#### **6.1 Kesimpulan**

#### **6.2 Saran**

# LAMPIRAN

## LAMPIRAN 1 : KODE SUMBER

### **admin\_useraddmaster**

Skrip ini diletakkan pada direktori `/usr/sesuatu` dan hanya dapat dieksekusi oleh *root*. Skrip ini berguna untuk menambahkan pengguna baru sesuai dengan konfigurasi baru yang telah ditetapkan.

**Kode 1:** Skrip menambahkan pengguna baru

```
#!/bin/csh -f
blah blah blah
blah blah blah
blah blah blah
blah blah blah
blah blah blah
```

### **getuser.cron**

Penjelasan skrip disini

**Kode 2:** *Cronjob* menambahkan pengguna baru

```
#!/bin/bash
# Change these two lines to localize to your system:
# Adapted from /usr/local/sbin/admin_useradd

cat /dev/null > $userlist
for (( i=0; i<${#listmailto[@]}; i++ ))
do
    uname=${listusername[$i]}
    mailto=${listmailto[$i]}

    echo "User $uname created, please use torqace wisely." | mail -s "Torqace
        user registration" $mailto
done
```



## LAMPIRAN 2 : BERKAS KONFIGURASI

### compute.xml

**Kode 3:** Berkas compute.xml

```
<?xml version="1.0" standalone="no"?>
<kickstart>
<description>
  Compute node XML file
</description>
</kickstart>
```

## LAMPIRAN 8 : UAT DAN KUESIONER

**Tabel 1:** Tabel UAT dan Kuesioner

No.	Langkah Penggunaan	Fitur Berjalan	Tingkat Kemudahan (1-5)	Tingkat Kepuasan (1-5)	Saran / Komentar
		Berhasil /Tidak	1:Sangat sulit ; 5:sangat mudah	1 : Sangat kecewa ; 5 : sangat puas	
Use Case : Login					
1.1	Pengguna berada pada halaman depan torqace				
1.2	Pengguna memasukkan username dan password pada field yang telah disediakan.Kemudian menekan tombol 'login'				
1.3	Apabila Sukses, maka pengguna masuk ke dalam sistem dan dihadapkan pada menu utama				
Use Case : Register					
2.1	Pengguna berada pada halaman registrasi pengguna torqace				

2.2	Pengguna memasukkan user-name,password, dan email pada field yang telah disediakan. Kemudian menekan tombol 'submit'				
2.3	Sistem akan mengonfirmasi masukan, dan akan mengirimkan email untuk memberitahu pengguna apabila proses pendaftaran telah selesai				
Use Case : Logout					
3.1	Pengguna memilih menu untuk melakukan logout				
3.2	Sistem akan mengeluarkan pengguna, dan pengguna tidak dapat menggunakan fitur-fitur utama aplikasi				
Use Case : Upload Job Sederhana					
4.1	Pengguna memilih menu upload file/project pada menu utama				
4.2	Pengguna memilih pilihan 'single file' pada tipe project				

4.3	Pengguna memilih berkas yang akan diunggah, mengisi label, dan menentukan apakah akan menimpa project sebelumnya dengan nama yang sama atau tidak				
4.4	Pengguna menekan tombol 'submit' dan mengonfirmasi				
4.5	Sistem akan menampilkan informasi terkait berkas yang diupload				
Use Case : Upload Job Compressed					
5.1	Pengguna memilih menu upload file/project pada menu utama				
5.2	Pengguna memilih pilihan 'compressed files' pada tipe project				
5.3	Pengguna memilih arsip yang akan diunggah, mengisi label, menentukan akan melakukan make atau tidak dan menentukan apakah akan menimpa project sebelumnya dengan nama yang sama atau tidak				
5.4	Pengguna menekan tombol 'submit' dan mengonfirmasi				

5.5	Sistem akan menampilkan informasi terkait berkas yang diupload dan diekstrak. Keluaran make juga akan ditampilkan bila dipilih				
Use Case : Upload Array Job					
6.1	Pengguna memilih menu upload file/project pada menu utama				
6.2	Pengguna memilih pilihan 'array' pada tipe project				
6.3	Pengguna memilih arsip-arsip yang akan diunggah, mengisi label, menentukan akan melakukan make atau tidak dan menentukan apakah akan menimpa project sebelumnya dengan nama yang sama atau tidak				
6.4	Pengguna menekan tombol 'submit' dan mengonfirmasi				
6.5	Sistem akan menampilkan informasi terkait berkas yang diupload dan diekstrak. Keluaran make juga akan ditampilkan bila dipilih				

Use Case : Melihat antrian pada queue					
7.1	Pengguna memilih menu queue status pada menu utama				
7.2	Pengguna berada pada halaman yang berisi informasi queue				
Use Case : Melihat detil antrian					
8.1	Dari halaman status queue, pengguna memilih job tertentu				
8.2	Informasi mengenai detil job tersebut ditampilkan dalam bentuk tabel				
8.2.1	Apabila job tersebut bukan milik pengguna, maka sistem akan melarang pengguna melihat informasi detil suatu job				
Use Case : Membuat script job					
9.1	Pengguna memilih untuk melakukan 'generate script' baik dari laporan upload berkas, atau dari penjelajahan direktori				
9.2	Pengguna mengisi nama job, parameter job, dan script yang akan dijalankan.				
9.3	Pengguna mengonfirmasi konfirmasi submit job				

9.4	Pengguna dapat melihat informasi script secara keseluruhan dan pesan apakah terjadi kegagalan atau tidak, serta id job yang diberikan				
Use Case : Load spesifikasi job lain					
10.1	Pengguna berada pada halaman untuk membuat script				
10.2	Pengguna memilih 'Load a Previous Job'				
10.3	Pengguna memilih job mana yang akan dimuat dan menekan tombol 'Load'				
10.4	Pengguna kembali ke halaman pembuatan script dengan spesifikasi job sebelumnya				
Use Case : Menjelajah Direktori					
11.1	Pengguna memilih menu 'View File/Project' pada menu utama				
11.2	Pengguna dapat melakukan navigasi untuk masuk ke dalam direktori tertentu, atau kembali ke direktori di atasnya, dan dapat melihat terdapat berkas apa saja dalam direktori				

Use Case : Menghapus Berkas/Direktori					
12.1	Pengguna berada pada halaman penjelajahan direktori				
12.2	Pengguna memilih pilihan untuk menghapus berkas/direktori di samping item yang akan dihapus				
12.3	Pengguna mengonfirmasi konfirmasi penghapusan				
Use Case : Mengunduh Berkas/Direktori					
13.1	Pengguna berada pada halaman penjelajahan direktori				
13.2	Pengguna memilih pilihan untuk mengunduh berkas/direktori di samping item yang akan dihapus				
Use Case : Melihat Berkas					
14.1	Pengguna berada pada halaman penjelajahan direktori				
14.2	Pengguna memilih berkas yang berupa berkas teks				
14.3	Sistem akan menampilkan konten dari berkas tersebut				