

# Mengenal User Mode Linux, Kernel dalam Kernel

**Abdullah Eli**  
eljabbar@yahoo.com

## ***Lisensi Dokumen:***

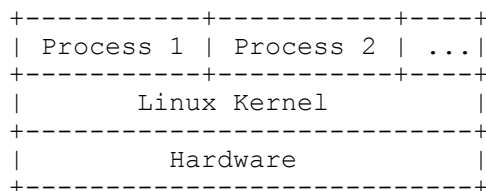
*Copyright © 2004 IlmuKomputer.Com*

*Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.*

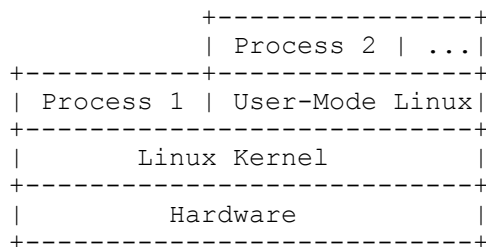
## **1.1 Pendahuluan**

User Mode Linux (UML) merupakan suatu mesin virtual yang dapat digunakan untuk menjalankan proses pada Linux secara aman tanpa beresiko merusak lingkungan Linux yang sebenarnya. UML secara umum bisa disebut sebagai kernel dalam kernel. Kernel dalam UML benar-benar mirip dengan kernel yang menjalankan sebuah sistem operasi linux. Bisa diubah kodenya, dcompile, distart, direstart, dan dishutdown.

Kernel linux yang biasa berjalan dan berkomunikasi di atas hardware PC. Strukturnya bisa digambarkan sebagai berikut:



Sedangkan UML saat dijalankan akan menciptakan menciptakan sebuah proses untuk instancenya sendiri-sendiri di atas Kernel linux yang asli. Strukturnya hampir sama, yaitu:



Pada sebuah UML dimungkinkan untuk ditambahkan resource hardware ataupun software secara virtual. Penyimpanan dari virtual mesin tersebut hanya berupa satu file tunggal untuk satu mesin virtual.

Berikut adalah beberapa kegunaan dari User Mode Linux:

1. Virtual hosting
2. Kernel development dan debugging

3. Process debugging
4. Secara aman mencoba kernel linux terbaru
5. Mencoba distribusi terbaru linux
6. Jaringan virtual
7. Disaster recovery practice, dan sebagainya

Berikut adalah hardware yang disupport UML:

1. Block device –pada UML digunakan satu file pada host yang berisikan file system. File ini akan menjadi block device di UML dan akan dimount seperti halnya file system pada disk fisik seperti pada kernel umumnya
2. Console dan serial lines – merupakan interface host termasuk file descriptor, ptys, ttys, ptsdevices, dan xterm
3. Network devices, SCSI devices, sound devices, PCI devices, dan USB devices

Alasan-alasan mengapa kita perlu memakai UML:

Jika User Mode Linux mengalami crash, kernel yang asli akan tetap baik-baik saja.

1. Kita bisa menjalankan sebuah UML meski tidak menggunakan login root.
2. Kita bisa melakukan debug terhadap UML sama seperti proses-proses lainnya.
3. Kita bisa menjalankan gprof (profiling) dan gcov (coverage testing).
4. Kita bisa bermain-main kernel tanpa merusak apapun.
5. Kita bisa menggunakannya untuk mencoba aplikasi-aplikasi baru.
6. Kita bisa mencoba mengembangkan sebuah kernel baru secara aman.
7. Kita bisa menjalankan banyak distro yang berlainan secara bersamaan.

## 1.2 Instalasi UML

UML tidak perlu diinstall dan bisa langsung dijalankan bahkan oleh seorang user biasa.

## 1.3 Fungsi-Fungsi dasar UML

### Menjalankan UML

Konsepnya adalah membuat salinan dari sebuah kernel UML untuk kemudian dimount agar bisa berjalan di dalam sistem. Kernel UML biasanya disebut 'root\_fs'.

Caranya adalah dengan menjalankan perintah: 'linux', UML akan mencoba mount file 'root\_fs' pada direktori yang sedang aktif. Terdapat beberapa parameter yang bias ditambahkan ketika menjalankan perintah tersebut, misalnya:

```
$linux umid=coba ubd0=root_fs_saya
```

Perintah tersebut akan menjalankan UML dengan id coba, dan file system yang digunakan adalah root\_fs\_saya.

### Login ke dalam UML

Secara default, terdapat account root dengan password "root" dan account user dengan password "user". Terdapat beberapa cara untuk login:

1. pada virtual console
2. melalui serial line
3. melalui jaringan

Untuk mengeset console ataupun serial line, perlu ditambahkan parameter ketika kita menjalankan UML. Format parameternya yaitu: device=channel. Device dispesifikasikan dengan "con" (console) atau "ssl" (serial line).

### Management Console

Management console (mconsole) merupakan interface low level ke kernel. Terdapat 3 hal yang dapat dilakukan dengan interface mconsole ini:

1. mendapatkan versi kernel
2. menambah dan membuang device

### 3. halt dan reboot

Untuk menjalankan console, dipanggil perintah: **uml\_mconsole nama\_id\_umid**, berikut adalah contoh beberapa perintah dalam mconsole:

- **version** –akan menampilkan versi UML
- **halt** atau **reboot** –akan mereboot atau halt UML
- **config** –untuk menambahkan device pada mesin virtual. Misalnya:  
**config eth0=TUNTAP,,,192.168.1.4** akan menambahkan device eth0 pada mesin virtual dengan tipe transport TUN/TAP.
- **remove** –untuk remove device, misalnya: **remove eth0** akan meremove device eth0 pada mesin virtual. Device yang diremove harus tidak aktif (tidak dimount).

### Setting Jaringan

Yang akan dilakukan adalah membuat virtual network. Pada masing-masing virtual host harus mempunyai network device, 1 untuk host biasa dan 2 atau lebih untuk router.

Ada 5 jenis transport yang tersedia dalam virtual machine UML untuk tukar-menukar paket data dengan host lain, yaitu:

#### - *Ethertap*

Format perintahnya secara umum:

**eth<n>=ethertap, <device>, <ethernet address>, <tap IP address>**

Contoh:

**eth0=ethertap,tap0,fe:fd:0:0:0:1,192.168.0.254**

→ artinya: meng-attach device eth0 ke host /dev/tap0, ethernet addressnya di-set fe:fd:0:0:0:1, dan IP addressnya di-set 192.168.0.254.

#### - *TUN/TAP*

Format perintahnya secara umum:

**eth<n>=tuntap, , , <IP address>**

Contoh:

**eth0=tuntap, , ,192.168.0.254**

#### - *Multicast*

Format perintahnya secara umum:

**eth<n>=mcast, <ethernet address>, <multicast address>, <multicast port>,  
<ttl>**

#### - *Switch daemon*

Format perintahnya secara umum:

**eth<n>=daemon, <ethernet address>, <socket type>, <control socket>,  
<data socket>**

#### - *Slip*

Format perintahnya secara umum:

**eth<n>=slip, <slip IP>**

Dari kelima jenis transport tersebut, yang memungkinkan mesin virtual untuk berkomunikasi dengan host adalah dengan menggunakan ethertap, TUN/TAP dan slip. Sedangkan daemon dan multicast menyediakan jaringan virtual untuk mesin virtual lainnya.

Untuk menambahkan device jaringan, ketika menjalankan UML dapat ditambahkan parameter sbb:  
**eth<n> = <transport>, <transport args>** misalnya:

**eth0 = ethertap, tap0, fe:fd:0:0:0:1, 192.168.0.254**

Parameter tersebut akan mengeset eth0 pada mesin virtual, dan /dev/tap0 pada host. IP address yang ditambahkan (192.168.0.254) harus berbeda dari IP yang diset pada mesin virtual.

Setelah network device dideskripsikan, UML harus di-boot dan harus dilakukan log in. Kemudian dilakukan pengaktifan interface, dengan format perintah:

```
ifconfig eth<n> <ip-address> up
```

Selanjutnya, di-set default route ke host, dengan format perintah:

```
route add default gw <host ip>
```

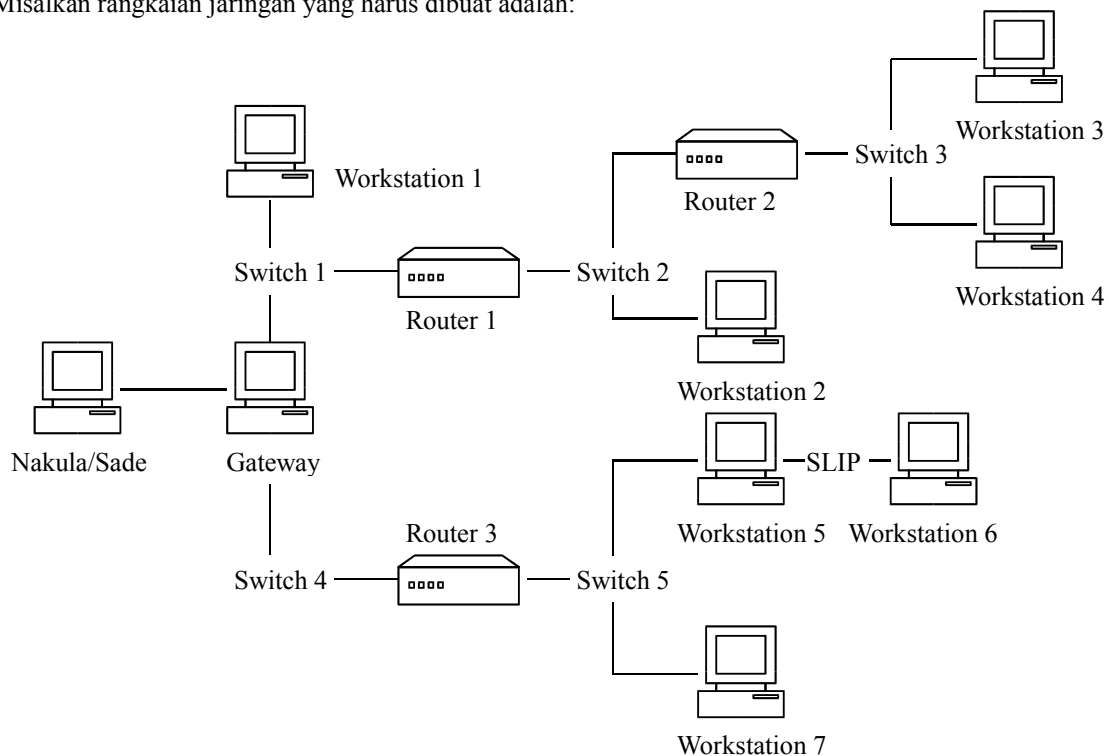
## 1.4 Kesimpulan

Beberapa contoh aplikasi User Mode Linux yang pernah dan berhasil dicoba oleh penulis adalah untuk:

1. Simulasi sebuah Local Area Network yang meliputi host, switch, dan router.
2. Instalasi DNS server dan Qmail server, serta menghubungkannya dengan internet.
3. Menggunakan layanan ftp, telnet, ssh, dan http lewat virtual host
4. Penerapan firewalling, meliputi service blocking, port forwarding, masquerading, dan virtual server.

## 1.5 Contoh implementasi, Simulasi Jaringan

Misalkan rangkaian jaringan yang harus dibuat adalah:



Keterangan:

1. Nakula/Sadewa berperan sebagai ISP dan memberikan IP 192.168.1.0/24 ke gateway.
2. Gateway mempunyai 3 network interface. Semua komputer dalam LAN kecuali Nakula/Sadewa harus mempunyai alamat network 192.168.1.0/24.
3. Hubungan antar komputer melalui switch, kecuali Nakula/Sadewa dan Workstation 5 dan 6.
4. Workstation 5 dan 6 hubungannya bersifat point-to-point. Implementasinya menggunakan switch dan dianggap 1 subnet dengan workstation 7.

Tabel Routing

Karena dalam jaringan tersebut terdiri dari 5 subnet, maka input mask yang dibutuhkan adalah 11111111.11111111.11111111.11100000 (255.255.255.224), dimana tiap subnet terdiri dari 30 node.

Tabel IP untuk tiap subnet adalah sebagai berikut:

Subnet bits	Network No.	Node Address		Broadcast Address
		From	To	
000	192.168.1.0	Reserved		
001	192.168.1.32	192.168.1.33	192.168.1.62	192.168.1.63
010	192.168.1.64	192.168.1.65	192.168.1.94	192.168.1.95

011	192.168.1.96	192.168.1.97	192.168.1.126	192.168.1.127
100	192.168.1.128	192.168.1.129	192.168.1.158	192.168.1.159
101	192.168.1.160	192.168.1.161	192.168.1.190	192.168.1.191
110	192.168.1.192	192.168.1.193	192.168.1.222	192.168.1.223
111	192.168.1.224	Reserved		

## Implementasi

- Langkah pertama yang dilakukan adalah membuat switch.

```
uml_switch -unix 15001 > /dev/null < /dev/null &
uml_switch -unix 16001 > /dev/null < /dev/null &
uml_switch -unix 17001 > /dev/null < /dev/null &
uml_switch -unix 18001 > /dev/null < /dev/null &
uml_switch -unix 19001 > /dev/null < /dev/null &
uml_switch -unix 20001 > /dev/null < /dev/null &
```

- Langkah kedua adalah membuat gateway, dengan eth0 ke 10.126.13.201, eth1 ke switch 15001, dan eth2 ke switch 18001.

```
linux ubd0=cow_gw,root_fs umid=gw con=pts con0=xterm eth0=tuntap,,,10.126.13.201
eth1=daemon,,,15001 eth2=daemon,,,18001 &
```

Setelah ini maka akan dibuat suatu virtual console baru untuk **gateway**. Setelah selesai akan muncul prompt untuk memasukkan login

uml@nakula login :

Masukkan login **root**.

- Langkah berikutnya adalah membuat workstation. Workstation hanya memiliki satu interface (eth), yaitu ke switch.

- ☞ workstation 1 ke switch 1 (15001)
- ☞ workstation 2 ke switch 2 (16001)
- ☞ workstation 3 ke switch 3 (17001)
- ☞ workstation 4 ke switch 3 (17001)
- ☞ workstation 5 ke switch 5 (19001)
- ☞ workstation 7 ke switch 5 (19001)

```
linux ubd0=cow_ws1,root_fs umid=ws1 con=pts con0=xterm eth0=daemon,,,15001 &
linux ubd0=cow_ws2,root_fs umid=ws2 con=pts con0=xterm eth0=daemon,,,16001 &
linux ubd0=cow_ws3,root_fs umid=ws3 con=pts con0=xterm eth0=daemon,,,17001 &
linux ubd0=cow_ws4,root_fs umid=ws4 con=pts con0=xterm eth0=daemon,,,17001 &
linux ubd0=cow_ws5,root_fs umid=ws5 con=pts con0=xterm eth0=daemon,,,19001 &
linux ubd0=cow_ws6,root_fs umid=ws6 con=pts con0=xterm eth0=daemon,,,20001 &
linux ubd0=cow_ws7,root_fs umid=ws7 con=pts con0=xterm eth0=daemon,,,19001 &
```

Sama seperti gateway, uml akan membuat virtual console baru untuk tiap workstation. Setelah itu user akan diminta memasukkan login untuk tiap virtual wks.

uml@nakula login :

Masukkan login **root**.

- Langkah berikutnya adalah membuat router. Router memiliki dua buah interface (eth)

- ☞ router 1 memiliki interface(eth0) ke switch 1 dan interface (eth1) ke switch 2
- ☞ router 2 memiliki interface(eth0) ke switch 2 dan interface (eth1) ke switch 3
- ☞ router 1 memiliki interface(eth0) ke switch 4 dan interface (eth1) ke switch 5

```
linux ubd0=cow_r1,root_fs umid=r1 con=pts con0=xterm eth0=daemon,,,15001
eth1=daemon,,,16001&
linux ubd0=cow_r2,root_fs umid=r2 con=pts con0=xterm eth0=daemon,,,16001
eth1=daemon,,,17001&
linux ubd0=cow_r3,root_fs umid=r3 con=pts con0=xterm eth0=daemon,,,18001
eth1=daemon,,,19001&
```

Sama seperti gateway, uml akan membuat virtual console baru untuk tiap router. Setelah itu user

akan diminta memasukkan login untuk tiap virtual router.  
uml@nakula login :  
Masukkan login **root**.

5. Lakukan **ifconfig** untuk setiap virtual console.

6. ifconfig untuk gateway

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 10.126.13.202 netmask 255.255.255.0 broadcast 10.126.13.255
ifconfig eth1 192.168.1.34 netmask 255.255.255.224 broadcast 192.168.1.63
ifconfig eth2 192.168.1.130 netmask 255.255.255.224 broadcast 192.168.1.159
```

7. ifconfig untuk router

- a. router 1

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.60 netmask 255.255.255.224 broadcast 192.168.1.63
ifconfig eth1 192.168.1.66 netmask 255.255.255.224 broadcast 192.168.1.95
```

- b. router 2

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.90 netmask 255.255.255.224 broadcast 192.168.1.95
ifconfig eth1 192.168.1.98 netmask 255.255.255.224 broadcast 192.168.1.127
```

- c. router 3

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.150 netmask 255.255.255.224 broadcast 192.168.1.159
ifconfig eth1 192.168.1.162 netmask 255.255.255.224 broadcast 192.168.1.191
```

8. ifconfig untuk workstation

- a. workstation 1

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.35 netmask 255.255.255.224 broadcast 192.168.1.63
```

- b. workstation 2

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.91 netmask 255.255.255.224 broadcast 192.168.1.95
```

- c. workstation 3

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.100 netmask 255.255.255.224 broadcast 192.168.1.127
```

- d. workstation 4

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.102 netmask 255.255.255.224 broadcast 192.168.1.127
```

- e. workstation 5

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.185 netmask 255.255.255.224 broadcast 192.168.1.191
```

- f. workstation 7

```
ifconfig lo 127.0.0.1 up
ifconfig eth0 192.168.1.188 netmask 255.255.255.224 broadcast 192.168.1.191
```

9. Langkah berikutnya adalah membuat routing untuk setiap virtual console

10. Untuk gateway

```
route add -net 192.168.1.64 netmask 255.255.255.224 gw 192.168.1.34
route add -net 192.168.1.96 netmaks 255.255.255.224 gw 192.168.1.98
route add -net 192.168.1.96 netmask 255.255.255.224 gw 192.168.1.34
```

```
route add -net 192.168.1.160 netmask 255.255.255.224 gw 192.168.1.130
route add -net 192.168.1.160 netmask 255.255.255.224 gw 192.168.1.162
```

11. Untuk router

a. router 1

```
route add -net 192.168.1.96 netmask 255.255.255.224 gw 192.168.1.66
route add default gw 192.168.3.34
```

b. router 2

```
route add default gw 192.168.1.66
```

c. router 3

```
route add default gw 192.168.1.130
```

12. Untuk workstation

a. workstation 1

```
route add default gw 192.168.1.34
```

b. workstation 2

```
route add default gw 192.168.1.66
```

c. workstation 3

```
route add default gw 192.168.1.66
```

d. workstation 4

```
route add default gw 192.168.1.66
```

e. workstation 5

```
route add default gw 192.168.1.162
```

f. workstation 7

```
route add default gw 192.168.1.162
```

13. Langkah berikutnya adalah mengecek routing table

a. untuk gateway

destination	gateway	genmask	interface
192.168.1.32	0.0.0.0	255.255.255.224	eth1
192.168.1.96	192.168.1.34	255.255.255.224	eth1
192.168.1.96	192.168.1.98	255.255.255.224	eth1
192.168.1.64	0.0.0.0	255.255.255.224	eth2
192.168.1.160	192.168.1.130	255.255.255.224	eth2
192.168.13.0	0.0.0.0	255.255.255.0	eth0
0.0.0.0	192.168.1.194	0.0.0.0	eth0

b. untuk router 1

destination	gateway	genmask	interface
192.168.1.32	0.0.0.0	255.255.255.224	eth0
192.168.1.96	0.0.0.0	255.255.255.224	eth1
192.168.1.96	192.168.1.98	255.255.255.224	eth1
0.0.0.0	192.168.3.34	0.0.0.0	eth0

c. untuk router 2

destination	gateway	genmask	interface
192.168.1.64	0.0.0.0	255.255.255.224	eth0
192.168.1.96	0.0.0.0	255.255.255.224	eth1

0.0.0.0	192.168.3.66	0.0.0.0	eth0
---------	--------------	---------	------

d. untuk router 3

destination	gateway	genmask	interface
192.168.1.32	0.0.0.0	255.255.255.224	eth0
192.168.1.128	0.0.0.0	255.255.255.224	eth1
0.0.0.0	192.168.1.130	0.0.0.0	eth0

e. untuk workstation 1

destination	gateway	genmask	interface
192.168.1.32	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.35	0.0.0.0	eth0

f. untuk workstation 2

destination	gateway	genmask	interface
192.168.1.64	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.91	0.0.0.0	eth0

g. untuk workstation 3

destination	gateway	genmask	interface
192.168.1.96	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.100	0.0.0.0	eth0

h. untuk workstation 4

destination	gateway	genmask	interface
192.168.1.96	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.102	0.0.0.0	eth0

i. untuk workstation 5

destination	gateway	genmask	interface
192.168.1.160	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.185	0.0.0.0	eth0

j. untuk workstation 7

destination	gateway	genmask	interface
192.168.1.160	0.0.0.0	255.255.255.224	eth0
0.0.0.0	192.168.1.188	0.0.0.0	eth0

14. Langkah terakhir adalah mengetes jaringan yang ada

Misal : workstation 1 (ip 192.168.1.35)

☞ Tes jaringan ke workstation 2

Ping 192.168.1.91

Jika berhasil maka akan muncul pesan :

Reply from 192.168.1.91: bytes=32 time<10ms TTL=253

☞ Tes jaringan ke workstation 3

Ping 192.168.1.100

Jika berhasil maka akan muncul pesan :

Reply from 192.168.1.100: bytes=32 time<10ms TTL=253

☞ Tes jaringan ke workstation 4

Ping 192.168.1.102

Jika berhasil maka akan muncul pesan :

Reply from 192.168.1.102: bytes=32 time<10ms TTL=253

☞ Tes jaringan ke workstation 5

Ping 192.168.1.185

Jika berhasil maka akan muncul pesan :

Reply from 192.168.1.185: bytes=32 time<10ms TTL=253

☞ Tes jaringan ke workstation 7



Ping 192.168.1.188  
Jika berhasil maka akan muncul pesan :  
Reply from 192.168.1.188: bytes=32 time<10ms TTL=253

### Ucapan terima kasih

1. Sdr. Chendra Hadi, rekan penulis yang dulu bersama-sama belajar tentang User Mode Linux dan menerapkannya pada praktikum jaringan komputer.
2. Sdr. Ade Reza Fernandez, untuk dokumentasinya yang baik karena dokumen saya sendiri sudah hilang.
3. Segenap pengurus dan komunitas Laboratorium AJK tempat penulis mengimplementasikan UML.

### Referensi

Selengkapnya bisa dibaca di <http://user-mode-linux.sourceforge.net> yang merupakan situs projectnya atau <http://usermodelinux.org> yang berupa sebuah web forum.

### BIOGRAFI PENULIS



Abdullah Eli, lahir di Mojokerto, 25 Juli 1981. Menamatkan SMU di SMUN 1 Sooko, Mojokerto pada tahun 1999. Saat ini sedang menyelesaikan Tugas Akhir / Skripsi program S1 jurusan Teknik Informatika di Institut Teknologi 10 Nopember, Surabaya. Bidang yang diminati adalah Software Engineering, Design Pattern, Object Oriented Application Design, Jaringan Komputer, Open Source Application Development dan Web Engineering.

Pernah menulis beberapa artikel di majalah Mikrodata dan tergabung dalam Mikrodata Forum. Pernah bekerja sebagai junior programmer di perusahaan konsultan manajemen selama 2 tahun. Selama 1 tahun ini bersama beberapa teman melakukan pengembangan dan pemasaran produk-produk software, baik itu yang dibangun sendiri maupun produk milik perusahaan lain. Aktif dalam beberapa organisasi mahasiswa (HMI Surabaya, 2000-2002), Lembaga Teknologi Mahasiswa Islam (LTMI, 2002-2003), dan tercatat sebagai anggota Microsoft User Group Indonesia (MUGI).

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:

Email : [eljabbar@yahoo.com](mailto:eljabbar@yahoo.com)

YM ID : eljabbar