BAB III
PEMBAHASAN

Sebagaimana yang telah disampaikan pada bab I, dalam kerja praktek ini dilakukan pembuatan aplikasi untuk melihat dan mengkonfigurasi *testbed* IMS yang terdapat di TELKOM RDC.

Pada bab ini akan dibahas analisis, perancangan sistem, implementasi dan evaluasi sistem. Analisis yang dilakukan meliputi analisis sistem yang sedang berjalan, analisis masalah, analisis kebutuhan fungsional, dan analisis kebutuhan non-fungsional. Perancangan sistem yang dilakukan meliputi perancangan antarmuka, modul dan algoritma. Implementasi meliputi implementasi perangkat lunak, implementasi perangkat keras, implementasi database dan implementasi antarmuka. Evaluasi sistem membahas mengenai kelebihan dan kekurangan dari aplikasi yang telah dibuat.

3.1 Analisis Sistem

Analisis sistem merupakan tahap yang dilakukan untuk mendapatkan pemahaman mengenai mapping tiap file, proses-proses yang terdapat pada sistem, keterkaitan antarproses, aliran data, pihak-pihak yang terlibat, hubungan sistem dengan lingkungan luar, dan hal lain yang berkaitan dengan sistem. Langkah-langkah yang dilakukan dalam tahap analisis sistem adalah analisis sistem yang sedang berjalan, analisis masalah, analisis kebutuhan fungsional, dan analisis kebutuhan non-fungsional.

3.1.1 Analisis Sistem yang sedang berjalan

Analisis terhadap sistem yang ada pada saat ini, dilakukan berdasarkan observasi dan interview dengan pembimbing yang ada di lapangan.

Analisis sistem yang dilakukan meliputi analisis prosedur konfigurasi *testbed* IMS, analisis data, dan analisis pengguna yang ada pada saat ini.

3.1.1.1 Analisis Prosedur

Prosedur untuk mengkonfigurasi *testbed* IMS adalah :

a) Administrator membuka file-file konfigurasi tersebut satu persatu dengan menggunakan text editor.
b) Administrator mencari *value* sesuai parameter yang terdapat di file konfigurasi *testbed* IMS.
c) Administrator menyimpan file-file konfigurasi yang telah diubah valuenya satu persatu.

3.1.1.2 Analisis Data

Terdapat 7 file konfigurasi *testbed* IMS, yakni pcscf.cfg, scscf.cfg, scscf.xml, icscf.cfg, icscf.xml, diameterpeerhss.xml, dan hss.properties. 7 file tersebut dapat dikelompokan menjadi :

a) P-CSCF : pcscf.cfg
b) S-CSCF : scscf.cfg, scscf.xml
c) I-CSCF : icscf.cfg, icscf.xml
d) HSS : diameterpeerhss.xml, hss.properties

Selain file-file tersebut, terdapat file konfigurasi lain yang terkait dengan file konfigurasi *testbed* IMS, diantaranya adalah file DNS. Untuk studi kasus di TELKOM RDC, file- file DNS tersebut adalah :

a) 8.5.4.2.2.2.6.e164.arpa.zone
b) 14.10.in-addr.arpa.zone
c) ims.telkom.co.id.zone
d) ims1.telkom.co.id.zone
e) ims2.telkom.co.id.zone
f) ims-ap.telkom.co.id.zone
g) ims-as.telkom.co.id.zone
h) ims-core.telkom.co.id.zone
i) named.conf
j) telkom.co.id.zone

Selain file-file tersebut dibuat 1 file konfigurasi baru untuk menopang kinerja aplikasi yang akan dibuat nantinya, yaitu ims1.cfg. File ini menunjukan letak direktori file-file konfigurasi yang ada di atas, yang kemudian akan digunakan dalam aplikasi saat pembacaan file-file tersebut.

Berdasarkan hasil diskusi, telah dipilih parameter-parameter penting yang terdapat pada file-file di atas. Parameter-parameter tersebut akan diakomodasi untuk keperluan aplikasi nantinya. Selain itu dari hasil diskusi diperoleh pula informasi bahwa terdapat relasi antara antara beberapa parameter di file-file tersebut, dimana parameter-parameter tersebut sebenarnya menunjukan sesuatu hal yang sama. Sebagai contoh parameter “alias” di file “icscf.cfg” memiliki kesamaan dengan parameter “FQDN” di file “icscf.xml”. Dengan begitu pengubahan value terhadap salah satu parameter tersebut sama artinya dengan mengubah value dari seluruh parameter yang identik. Berdasarkan kondisi ini, maka untuk keperluan aplikasi ini diperlukan pengelompokkan terhadap parameter-parameter identik tersebut. Selanjutnya, pada masing-masing kelompok akan ditentukan satu parameter sebagai acuan untuk mewakili kelompoknya. Nantinya, aplikasi cukup membaca value dari parameter acuan di masing-masing kelompok dan menampilkannya sebagai konfigurasi *testbed* IMS. Jika Admin mengubah value dari salah satu parameter, maka value dari seluruh parameter yang identik dalam kelompok yang sama secara otomatis aka berubah. Berikut kelompok-kelompok parameter identik yang berhasil di identifikasi:

1) Nama Domain

Kelompok ini menunjukkan domain dari konfigurasi *testbed* IMS. Untuk studi kasus di TELKOM RDC, domain *testbed* IMS yang ada diberi nama “ims.telkom.co.id”. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value “ims.telkom.co.id” di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **“Realm”** yg terdapat di file “scscf.xml”. Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
icscf.cfg	alias=ims.telkom.co.id modparam("icscf","orig_ioi","ims.telkom.co.id") modparam("icscf","term_ioi","ims.telkom.co.id") insert_hf_value("Route","<sip:term@mgcf.ims.telkom.co.id;lr>");
pcscf.cfg	modparam("pcscf","orig_ioi","ims.telkom.co.id") modparam("pcscf","term_ioi","ims.telkom.co.id") P_add_p_visited_network_id("ims.telkom.co.id"); P_access_network_info("ims.telkom.co.id");

scscf.cfg	S_assign_server_unreg("ims.telkom.co.id", "orig");
	S_assign_server_unreg("ims.telkom.co.id", "term");
	if (!S_is_integrity_protected("ims.telkom.co.id")){
	if (!S_is_authorized("ims.telkom.co.id")) {
	S_challenge("ims.telkom.co.id");
	if (S_assign_server("ims.telkom.co.id")){
	insert_hf_value("Route","<sip:term@mgcf.ims.telkom.co.id;lr>");

scscf.xml	*Realm="ims.telkom.co.id" <Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/>
icscf.xml	Realm="ims.telkom.co.id" <Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/> <Realm name="ims.telkom.co.id"> <Realm name="another.ims.telkom.co.id">
DiameterPeerHSS .xml	Realm="ims.telkom.co.id" <Peer FQDN="icscf.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868" /> <Peer FQDN="scscf.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868" />
8.5.4.2.2.2.6.e 164.arpa.zone	IN NS ns.ims.telkom.co.id. 1.1.0.0 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4580011@ims.telkom.co.id!" . ;1.0.0.0 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4580001@ims.telkom.co.id!" . 0.0.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581900@ims.telkom.co.id!" . 1.0.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581901@ims.telkom.co.id!" . 2.0.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581902@ims.telkom.co.id!" . 3.0.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581903@ims.telkom.co.id!" . 9.1.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581919@ims.telkom.co.id!" . 9.9.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581999@ims.telkom.co.id!" . 1.9.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581991@ims.telkom.co.id!" . 0.2.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581920@ims.telkom.co.id!" . 1.9.8.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581891@ims.telkom.co.id!" . 2.9.9.1 IN NAPTR 10 10 "u" "sip+E2U" "!^.*$!sip:4581992@ims.telkom.co.id!" .
14.10.in-addr.arpa.zone	IN NS ns.ims.telkom.co.id. 14.18 IN PTR trcf.ims.telkom.co.id. 16.18 IN PTR ims.telkom.co.id. 19.18 IN PTR mgcf.ims.telkom.co.id. 23.18 IN PTR presence.ims.telkom.co.id. 24.18 IN PTR rls.ims.telkom.co.id.

	25.18 IN PTR xdms.ims.telkom.co.id. 26.18 IN PTR sipsee.ims.telkom.co.id. 31.18 IN PTR anubis.ims.telkom.co.id. 32.18 IN PTR coms.ims.telkom.co.id. 33.18 IN PTR occas.ims.telkom.co.id.
ims.telkom.co.id.zone	$ORIGIN ims.telkom.co.id. ims.telkom.co.id. 1D IN A 10.14.18.16 ims.telkom.co.id. 1D IN NAPTR 10 50 "s" "SIP+D2U" "" _sip._udp ims.telkom.co.id. 1D IN NAPTR 20 50 "s" "SIP+D2T" "" _sip._tcp

2) Log Standar Error I-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan logging terhadap error yang terjadi di I-CSCF. Parameter ini dijumpai di file “icscf.cfg”, yaitu “log_stderror=yes”. Value “yes” menyatakan bahwa jika terjadi error di I-CSCF, maka error tersebut akan dicatat dalam log file (history), sedangkan “no” berarti error tidak dicatat.

Parameter “log_stderror” sendiri sebenarnya dapat dijumpai di file lain, seperti scscf.cfg, pcscf.cfg. Namun sesuai dengan hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	log_stderror=yes

3) Level Memory Log di I-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan *logging* terhadap error yang terjadi di I-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file **“**icscf.cfg**”** yaitu “memlog=3”. Parameter ini sendiri baru akan diperhitungkan, jika parameter “log_stderror” di file “icscf.cfg” mempunyai value “yes”.

Parameter “memlog” sendiri sebenarnya dapat dijumpai di file lain, seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter-parameter tersebut ternyata digunakan secara khusus untuk masing-masing

elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	**memlog** =5

4) SIP Warning I-CSCF

Parameter ini berisi pengaturan tentang perlu tidaknya dilakukan *logging* terhadap *anomaly* protocol SIP yang terjadi di I-CSCF. Parameter dapat ini dijumpai di file "icscf.cfg" yaitu "sip_warning=yes". Value "yes" berarti bahwa jika terjadi *anomaly* SIP di I-CSCF, maka anomaly tersebut akan dicatat dalam log file, sedangkan "no" berarti *anomaly* tidak dicatat.

Parameter "sip_warning" sendiri sebenarnya dapat dijumpai di file lain, seperti scscf.cfg, pscsf.cfg. Namun sesuai hasil observasi dan diskusi, parameter-parameter tersebut ternyata digunakan secara khusus untuk masing-masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	sip_warning=yes

5) Level debug di I-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan *debug* proses & aktivitas di I-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file "icscf.cfg" yaitu "debug=3".

Parameter "debug" sendiri sebenarnya dapat dijumpai di file lain, seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter-parameter tersebut ternyata digunakan secara khusus untuk masing-masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	debug=3

6) Forking di I-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan mekanisme *forking* di I-CSCF. Parameter ini dijumpai di file “icscf.cfg”, yaitu “fork=yes”. Value “yes” berarti bahwa *forking* diperbolehkan di I-CSCF, sedangkan “no” berarti *forking* tidak boleh dilakukan.

Parameter “fork” sendiri sebenarnya dapat dijumpai di file lain, seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter–parameter tersebut ternyata digunakan secara khusus untuk masing-masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	fork=yes

7) Level Children I-CSCF

Parameter ini berisi pengaturan mengenai tingkat *children* di I-CSCF. Value yg dapat dipilih berada di rentang 1-5. Parameter ini dijumpai di file “icscf.cfg” yaitu “children=4”. Parameter ini sendiri baru akan diperhitungkan, jika parameter “fork” di file “icscf.cfg” mempunyai value “yes”.

Parameter “children” sendiri sebenarnya dapat dijumpai di file lain,seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	children=4

8) Check Via SIP Header I-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan pengecekan *field* “via” yang ada di dalam message SIP yaitu di SIP header.

Parameter “check_via” sendiri sebenarnya dapat dijumpai di file lain,seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing

elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	check_via=no

9) Look – Up DNS I-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *look up* ke DNS. Parameter "dns" sendiri sebenarnya dapat dijumpai di file lain,seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	dns=no

10) Reversed Look-Up DNS I-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *reversed look up* ke DNS. Parameter "rev_dns" sendiri sebenarnya dapat dijumpai di file lain,seperti scscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
icscf.cfg	rev_dns=no

11) Nama I-CSCF

Kelompok ini menunjukkan nama atau alias dari I-CSCF. Untuk studi kasus di TELKOM RDC, I-CSCF *testbed* IMS yang ada diberi nama "icscf.ims.telkom.co.id". Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "icscf.ims.telkom.co.id" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"FQDN"** yg terdapat di file "icscf.xml".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
icscf.cfg	alias=icscf.ims.telkom.co.id advertised_address=icscf.ims.telkom.co.id modparam("icscf","name","icscf.ims.telkom.co.id")
icscf.xml	*FQDN="icscf.ims.telkom.co.id"
DiameterPeerHSS.xml	<Peer FQDN="icscf.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868" />

12) IP Address I-CSCF

Kelompok ini menunjukkan IP Address dari I-CSCF. Untuk studi kasus di TELKOM RDC, I-CSCF *testbed* IMS yang ada diberi IP address 10.14.18.16. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "10.14.18.16" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"listen"** yg terdapat di file "icscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
icscf.cfg	*listen=10.14.18.16 modparam("icscf","icid_gen_addr","10.14.18.16")
icscf.xml	<Acceptor port="3868" bind="10.14.18.16"/>
ims.telkom.co.id.zone	ims.telkom.co.id. 1D IN A 10.14.18.16 icscf 1D IN A 10.14.18.16

13) SIP Port I-CSCF

Kelompok ini menunjukkan alamat port, tempat dimana I-CSCF mendengarkan trafik SIP yang masuk. Untuk studi kasus di TELKOM RDC, I-CSCF *testbed* IMS yang ada akan mendengarkan incoming SIP trafik di port 5060. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "5060" yang berelasi dengan I-CSCF di semua file. Parameter yang dianggap sebagai acuan wakil dari kelompok ini adalah parameter **"port"** yg terdapat di file "icscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
icscf.cfg	*port=5060
ims.telkom.co.id.zone	_sip 1D SRV 0 0 5060 icscf _sip._udp 1D SRV 0 0 5060 icscf _sip._tcp 1D SRV 0 0 5060 icscf
Ims2.telkom.co.id.zone	_sip 1D SRV 0 0 5060 icscf _sip._udp 1D SRV 0 0 5060 icscf _sip._tcp 1D SRV 0 0 5060 icscf

14) Diameter Port I-CSCF

Kelompok ini menunjukkan alamat port, tempat dimana I-CSCF mendengarkan trafik diameter yang masuk. Untuk studi kasus di TELKOM RDC, I-CSCF *testbed* IMS yang ada akan mendengarkan incoming diameter trafik di port 3868. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value “3868” yang berelasi dengan I-CSCF di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **“Acceptor port”** yang terdapat di file “icscf.xml”.

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
icscf.xml	*<Acceptor port="3868" bind="10.14.18.16"/>
DiameterPeerHSS.xml	<Peer FQDN="icscf.ims.telkom.co.id" Realm= "ims.telkom.co.id" port="3868" />

15) Log Standar Error S-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan logging terhadap error yang terjadi di S-CSCF. Parameter ini dapat dijumpai di file “scscf.cfg”, yaitu “log_stderror=yes”. Value “yes” menyatakan bahwa jika terjadi error di S-CSCF, maka error tersebut akan dicatat dalam log file (history), sedangkan “no” berarti error tidak dicatat.

Parameter “log_stderror” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai dengan hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing -

masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	log_stderror=yes

16) Level Memory Log di S-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan *logging* terhadap error yang terjadi di S-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file **"**scscf.cfg**"** yaitu "memlog=3". Parameter ini sendiri baru akan diperhitungkan, jika parameter "log_stderror" di file "scscf.cfg" mempunyai value "yes".

Parameter "memlog" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	memlog =5

17) SIP Warning S-CSCF

Parameter ini berisi pengaturan tentang perlu tidaknya dilakukan *logging* terhadap *anomaly* protocol SIP yang terjadi di S-CSCF. Parameter dapat ini dijumpai di file "scscf.cfg" yaitu "sip_warning=yes". Value "yes" berarti bahwa jika terjadi *anomaly* SIP di S-CSCF, maka anomaly tersebut akan dicatat dalam log file, sedangkan "no" berarti *anomaly* tidak dicatat.

Parameter "sip_warning" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pscsf.cfg. Namun sesuai hasil observasi dan diskusi, parameter-parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	Sip_warning=yes

18) Level debug di S-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan *debug* proses dan aktivitas di S-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file "scscf.cfg" yaitu "debug=3".

Parameter "debug" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	debug=3

19) Forking di S-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan mekanisme *forking* di S-CSCF. Parameter ini dijumpai di file "scscf.cfg", yaitu "fork=yes". Value "yes" berarti bahwa *forking* diperbolehkan di S-CSCF, sedangkan "no" berarti *forking* tidak boleh dilakukan.

Parameter "fork" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	fork=yes

20) Level Children S-CSCF

Parameter ini berisi pengaturan mengenai tingkat *children* di S-CSCF. Value yg dapat dipilih berada di rentang 1-5. Parameter ini dijumpai di file "scscf.cfg" yaitu "children=4". Parameter ini sendiri baru akan diperhitungkan, jika parameter "fork" di file "scscf.cfg" mempunyai value "yes".

Parameter "children" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter –

parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	children=4

21) Check Via SIP Header S-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan pengecekan *field* "via" yang ada di dalam message SIP yaitu di SIP header.

Parameter "Check_via" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	check_via=no

22) Look – Up DNS S-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *look up* ke DNS. Parameter "dns" sendiri sebenarnya dapat dijumpai di file lain,seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	dns=no

23) Reversed Look-Up DNS S-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *reversed look up* ke DNS. Parameter "rev_dns" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, pcscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing –

masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
scscf.cfg	rev_dns=no

24) Nama S-CSCF

Kelompok ini menunjukkan nama atau alias dari S-CSCF. Untuk studi kasus di TELKOM RDC, S-CSCF *testbed* IMS yang ada diberi nama "scscf.ims.telkom.co.id". Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "scscf.ims.telkom.co.id" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"FQDN"** yg terdapat di file "scscf.xml".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.cfg	alias=scscf.ims.telkom.co.id advertised_address=scscf.ims.telkom.co.id modparam("scscf","name","sip:scscf.ims.telkom.co.id") modparam("isc","my_uri","scscf.ims.telkom.co.id")
scscf.xml	*FQDN="scscf.ims.telkom.co.id"
DiameterPeerHSS.xml	<PeerFQDN="scscf.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868" />
14.10.in-addr.arpa.zone	17.18 IN PTR scscf.ims.telkom.co.id.

25) IP Address S-CSCF

Kelompok ini menunjukkan IP Address dari S-CSCF. Untuk studi kasus di TELKOM RDC, S-CSCF *testbed* IMS yang ada diberi IP address "10.14.18.17". Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "10.14.18.17" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"listen"** yg terdapat di file "scscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.cfg	*listen=10.14.18.17 t_relay_to_udp("10.14.18.17",5060);
scscf.xml	<Acceptor port="3868" bind="10.14.18.17"/>
ims.telkom.co.id.zone	Scscf 1D IN A 10.14.18.17

26) SIP Port S-CSCF

Kelompok ini menunjukkan alamat port, tempat dimana S-CSCF mendengarkan trafik SIP yang masuk. Untuk studi kasus di TELKOM RDC, S-CSCF *testbed* IMS yang ada akan mendengarkan incoming SIP trafik di port 5060. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "5060" yang berelasi dengan S-CSCF di semua file. Parameter yang dianggap sebagai acuan wakil dari kelompok ini adalah parameter **"port"** yg terdapat di file "scscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.cfg	*port=5060 t_relay_to_udp("10.14.18.17",5060);
ims.telkom.co.id.zone	_sip.scscf 1D SRV 0 0 5060 scscf _sip._udp.scscf 1D SRV 0 0 5060 scscf _sip._tcp.scscf 1D SRV 0 0 5060 scscf

27) Diameter Port S-CSCF

Kelompok ini menunjukkan alamat port, tempat dimana S-CSCF mendengarkan trafik diameter yang masuk. Untuk studi kasus di TELKOM RDC, S-CSCF *testbed* IMS yang ada akan mendengarkan incoming diameter trafik di port 3868. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "3868" yang berelasi dengan S-CSCF di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"Acceptor port"** yang terdapat di file "scscf.xml".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.xml	*<Acceptor port="3868" bind="10.14.18.17"/>

DiameterPeerHSS.xml	<Peer FQDN="scscf.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868" />

28) Default Time Expiration of Registration

Parameter ini menunjukkan pengaturan mengenai berapa lama masa berlaku dari sebuah status registrasi *testbed* IMS per user dalam satuan milisecond. Parameter ini hanya dijumpai di file “scscf.cfg”, yaitu pada baris dengan pola “modparam("scscf","registration_default_expires",xxxx)”, di mana xxxx merupakan value dari expiration time tersebut.

Untuk studi kasus di TELKOM RDC, value dari parameter ini diset 3600, yang artinya masa berlaku dari sebuah status registrasi adalah selama 1 jam. Setelah 1 jam, status registrasi akan diperbaharui kembali untuk 1 jam berikutnya.

Nama File	Parameter
scscf. cfg	modparam("scscf","registration_default_expires",3600)

29) Default Algorithm for Registration

Parameter ini menunjukkan pengaturan tentang algoritma yang akan digunakan untuk melakukan authentikasi pada saat proses registrasi user *testbed* IMS. Pilihan algoritma ini dapat diset langsung di S-CSCF atau mengikuti algoritma yang telah diatur di HSS (HSS-Selected).

Parameter ini hanya dijumpai di file “scscf.cfg” pada baris-baris dengan pola “modparam("scscf"," registration_default_algorithm ",”xxxx”)”, di mana xxxx menunjukkan pilihan algoritma.

Nama File	Parameter
scscf. cfg	#modparam("scscf","registration_default_algorithm","AKAv1-MD5") #modparam("scscf","registration_default_algorithm","AKAv2-MD5") #modparam("scscf","registration_default_algorithm","MD5") #modparam("scscf","registration_default_algorithm","CableLabs-Digest") #modparam("scscf","registration_default_algorithm","TISPAN-HTTP_DIGEST_MD5") *modparam("scscf","registration_default_algorithm","HSS-Selected") #modparam("scscf","registration_default_algorithm","NASS-Bundled")

30) Log Standar Error P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan logging terhadap error yang terjadi di P-CSCF. Parameter ini dapat dijumpai di file

“pcscf.cfg”, yaitu “log_stderror=yes”. Value “yes” menyatakan bahwa jika terjadi error di P-CSCF, maka error tersebut akan dicatat dalam log file (history), sedangkan “no” berarti error tidak dicatat.

Parameter “log_stderror” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai dengan hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	log_stderror=yes

31) Level Memory Log di P-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan *logging* terhadap error yang terjadi di P-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file **“**pcscf.cfg**”** yaitu “memlog=5”. Parameter ini sendiri baru akan diperhitungkan, jika parameter “log_stderror” di file “pcscf.cfg” mempunyai value “yes”.

Parameter “memlog” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	memlog =5

32) SIP Warning P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *logging* terhadap *anomaly* protocol SIP yang terjadi di P-CSCF. Parameter dapat ini dijumpai di file “pcscf.cfg” yaitu “sip_warning=yes”. Value “yes” berarti bahwa jika terjadi *anomaly* SIP di P-CSCF, maka anomaly tersebut akan dicatat dalam log file, sedangkan “no” berarti *anomaly* tidak dicatat.

Parameter “sip_warning” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter-parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	Sip_warning=yes

33) Level Debug di P-CSCF

Parameter ini berisi pengaturan mengenai tingkat kedetilan debug proses dan aktivitas di P-CSCF. Value yg dapat dipilih berada di rentang 1-5, dimana 1 berarti tidak detil, sedangkan 5 berarti sangat detil. Parameter ini dijumpai di file “pcscf.cfg” yaitu “debug=3”.

Parameter “debug” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter - parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	debug=3

34) Forking di P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan mekanisme *forking* di P-CSCF. Parameter ini dijumpai di file “pcscf.cfg”, yaitu “fork=yes”. Value “yes” berarti bahwa *forking* diperbolehkan di P-CSCF, sedangkan “no” berarti *forking* tidak boleh dilakukan.

Parameter “fork” sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing - masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	fork=yes

35) Level Children P-CSCF

Parameter ini berisi pengaturan mengenai tingkat *children* di P-CSCF. Value yg dapat dipilih berada di rentang 1-5. Parameter ini dijumpai di file "pcscf.cfg" yaitu "children=3". Parameter ini sendiri baru akan diperhitungkan, jika parameter "fork" di file "pcscf.cfg" mempunyai value "yes".

Parameter "children" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	children=4

36) Check Via SIP Header P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan pengecekan *field* "via" yang ada di dalam message SIP yaitu di SIP header.

Parameter "check_via" sendiri sebenarnya dapat dijumpai di file lain,seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	check_via=no

37) Look – Up DNS P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *look up* ke DNS. Parameter "dns" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	dns=no

38) Reversed Look-Up DNS P-CSCF

Parameter ini berisi pengaturan mengenai perlu tidaknya dilakukan *reversed look up* ke DNS. Parameter "rev_dns" sendiri sebenarnya dapat dijumpai di file lain, seperti icscf.cfg, scscf.cfg. Namun sesuai hasil observasi dan diskusi, parameter – parameter tersebut ternyata digunakan secara khusus untuk masing – masing elemen (tidak identik dan tidak saling berelasi satu sama lain), sehingga tidak dapat dijadikan satu kelompok.

Nama File	Parameter
pcscf.cfg	rev_dns=no

39) Nama P-CSCF

Kelompok ini menunjukkan nama atau alias dari P-CSCF. Untuk studi kasus di TELKOM RDC, P-CSCF *testbed* IMS yang ada diberi nama "pcscf.ims.telkom.co.id". Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "pcscf.ims.telkom.co.id" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"alias"** yg terdapat di file "pcscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
pcscf.cfg	*alias=pcscf.ims.telkom.co.id advertised_address=pcscf.ims.telkom.co.id modparam("pcscf","name","sip:pcscf.ims.telkom.co.id")
14.10.in-addr.arpa.zone	15.18 IN PTR pcscf.ims.telkom.co.id.

40) IP Address P-CSCF

Kelompok ini menunjukkan IP Address dari P-CSCF. Untuk studi kasus di TELKOM RDC, P-CSCF *testbed* IMS yang ada diberi IP address "10.14.18.15". Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "10.14.18.15" di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"listen"** yg terdapat di file "pcscf.cfg".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
pcscf.cfg	*listen=10.14.18.15 modparam("pcscf","ipsec_host","10.14.18.15") modparam("pcscf","icid_gen_addr","10.14.18.15")
ims.telkom.co.id.zone	pcscf 1D IN A 10.14.18.15

41) SIP Port P-CSCF

Kelompok ini menunjukkan alamat port, tempat dimana P-CSCF mendengarkan trafik SIP yang masuk. Untuk studi kasus di TELKOM RDC, P-CSCF *testbed* IMS yang ada akan mendengarkan incoming SIP trafik di port 5060. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value “5060” yang berelasi dengan P-CSCF di semua file. Parameter yang dianggap sebagai acuan wakil dari kelompok ini adalah parameter **“port”** yg terdapat di file “pcscf.cfg”.

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
pcscf.cfg	*port=5060 modparam("pcscf","ipsec_port_c",5060) modparam("pcscf","ipsec_port_s",5060)
ims.telkom.co.id.zone	_sip.pcscf 1D SRV 0 0 5060 pcscf _sip._udp.pcscf 1D SRV 0 0 5060 pcscf _sip._tcp.pcscf 1D SRV 0 0 5060 pcscf

42) RTP Proxy

Kelompok ini menunjukkan pengaturan apakah RTP Proxy akan diaktifkan, dan jika ya, bagaimana settingnya (protocol, IP address dan portnya). Parameter ini hanya dijumpai di file “pcscf.cfg” yaitu pada baris-baris dengan pola “modparam("pcscf","rtpproxy_enable", [0/1])” dan “modparam("pcscf","rtpproxy_socket", "<protocol>:<ip_address>:<port>")”.

a. Baris dengan pola “modparam("pcscf","rtpproxy_enable",[0/1])” menunjukkan apakah RTP Proxy diaktifkan atau tidak, di mana 0 berarti RTP Proxy tidak diaktifkan dan 1 berarti RTP Proxy diaktifkan.

b. Baris dengan pola “modparam("pcscf","rtpproxy_socket", "<protocol>:<ip_address>:<port>")” menunjukkan setting protocol, IP address dan port dari RTP Proxy. Setting ini hanya akan diperhitungkan jika RTP Proxy diaktifkan.

Untuk studi kasus di TELKOM RDC, RTP Proxy diaktifkan. Ini terlihat pada baris yang berbunyi “modparam("pcscf","rtpproxy_enable", 1) “. Setting dari RTP Proxy ini dijumpai di baris lain,yaitu “modparam("pcscf","rtpproxy_socket", "udp:127.0.0.1:34999")” yang berarti RTP Proxy menggunakan protocol UDP dan aktif di IP address 127.0.0.1 di port 34999.

Nama File	Parameter
pcscf.cfg	modparam("pcscf","rtpproxy_socket", "udp:127.0.0.1:34999") modparam("pcscf","rtpproxy_enable", 1)

43) TLS

Kelompok ini menunjukkan pengaturan apakah TLS akan diaktifkan dan jika ya, bagaimana settingnya (IP address dan portnya). Parameter ini hanya dijumpai di file “pcscf.cfg” yaitu pada baris-baris dengan pola “enable_tls=[yes/no]”,“modparam("pcscf","use_tls",[0/1])”,“listen=tls:<ip_address>”, tls_port_no=<port>

a. Baris dengan pola “enable_tls=[yes/no]” dan “modparam("pcscf","use_tls",[0/1])” menunjukkan apakah TLS diaktifkan atau tidak, di mana “no” dan 0 berarti TLS tidak diaktifkan dan “yes” dan 1 berarti TLS diaktifkan.
b. Baris dengan pola “listen=tls:<ip_address>” menunjukkan setting IP address TLS dan “tls_port_no=<port>” menunjukkan port TLS. Setting ini hanya akan diperhitungkan jika TLS diaktifkan.

Untuk studi kasus di TELKOM RDC, TLS tidak diaktifkan.

Nama File	Parameter
pcscf.cfg	#listen=tls:10.14.18.15 #tls_port_no=4061 #enable_tls=yes *modparam("pcscf","use_tls",0)

44) HSS port

Kelompok ini menunjukan port dari web server, dikelompokkan berdasar value “80”.

Nama File	Parameter
hss.properties	port=80

45) Nama HSS

Kelompok ini menunjukkan nama atau alias dari HSS. Untuk studi kasus di TELKOM RDC, HSS *testbed* IMS yang ada diberi nama “hss.ims.telkom.co.id”. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value “hss.ims.telkom.co.id” di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **“FQDN”** yg terdapat di file “DiameterPeerHSS.xml”.

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.xml	<Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/> <DefaultRoute FQDN="hss.ims.telkom.co.id" metric="10"/>
icscf.xml	<Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/> <DefaultRoute FQDN="hss.ims.telkom.co.id" metric="10"/> <Route FQDN="hss.ims.telkom.co.id" metric="20"/>
DiameterPeerHSS.xml	*FQDN="hss.ims.telkom.co.id"/>
14.10.in-addr.arpa.zone	18.18 IN PTR hss.ims.telkom.co.id.

46) IP Address HSS

Kelompok ini menunjukkan IP Address dari HSS. Untuk studi di TELKOM RDC, HSS *testbed* IMS yang ada diberi IP address “10.14.18.18”. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value “10.14.18.18” di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **“host”** yg terdapat di file “hss.properties”

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
hss.properties	*host=10.14.18.18
DiameterPeerHSS.xml	<Acceptor port="3868" bind="10.14.18.18" />
ims.telkom.co.id.zone	hss 1D IN A 10.14.18.18

47) Diameter Port HSS

Kelompok ini menunjukkan alamat port, tempat dimana HSS mendengarkan trafik diameter yang masuk. Untuk studi kasus di TELKOM RDC Bandung, HSS IMS yang ada akan mendengarkan incoming Diameter trafik di port 3868. Oleh karena itu, proses pembentukan kelompok ini dilakukan dengan mencari parameter dengan value "3868" yang berelasi dengan HSS di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"Acceptor port"** yang terdapat di file "DiameterPeerHSS.xml"

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf. xml	<Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/>
icscf. xml	<Peer FQDN="hss.ims.telkom.co.id" Realm="ims.telkom.co.id" port="3868"/>
DiameterPeerHSS.xml	*<Acceptor port="3868" bind="10.14.18.18" />

48) Default Time Expiration of Subscription

Parameter ini menunjukkan pengaturan tentang berapa lama masa berlaku dari sebuah status subskripsi *testbed* IMS per user dalam satuan *milisecond*. Untuk studi kasus di TELKOM RDC, value dari parameter ini diset 3600, yang artinya masa berlaku dari sebuah subskripsi adalah selama 1 jam. Setelah 1 jam, status subskripsi akan diperbaharui kembali untuk 1 jam berikutnya.

Proses pembentukan kelompok ini dilakukan dengan mencari parameter yang mempunyai value "3600" yang berelasi dengan subskripsi di semua file. Parameter yang dianggap sebagai acuan atau wakil dari kelompok ini adalah parameter **"expiry_time_lim"** yg terdapat di file "hss.properties".

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut:

Nama File	Parameter
scscf.cfg	modparam("scscf","subscription_default_expires",3600)
hss.properties	*expiry_time_lim=3600

49) Default Time Expiration of Dialog

Parameter ini menunjukkan pengaturan tentang berapa lama masa berlaku dari sebuah dialog *testbed* IMS per user dalam satuan *milisecond.* Parameter ini dijumpai pada baris-baris dengan pola modparam("<elemen>",dialogs_expiration_time",<value>), di mana elemen dapat diisi dengan "pcscf" atau "scscf" (disesuaikan dengan filenya) dan value merupakan durasi dialog yang diinginkan.

Untuk studi kasus di TELKOM RDC, value dari parameter ini diset 3600, yang artinya masa berlaku dari sebuah dialog adalah selama 1 jam. Setelah 1 jam, dialog dapat dilakukkan kembali untuk 1 jam berikutnya.

Adapun parameter-parameter yang termasuk ke dalam kelompok ini adalah sebagai berikut :

Nama File	Parameter
scscf.cfg	modparam("scscf","dialogs_expiration_time",3600)
pcscf.cfg	modparam("pcscf","dialogs_expiration_time",3600)

3.1.1.3 Analisis Pengguna

Berdasarkan observasi dan diskusi dengan pembimbing yang ada di lapangan, pada saat ini file konfigurasi *testbed* IMS hanya dapat dilihat dan dikonfigurasi oleh administrator yang ada di TELKOM RDC. User selain administrator tidak dapat melihat dan mengkonfigurasi, meskipun untuk kepentingan belajar.

3.1.2 Analisis Masalah

Teknik konfigurasi *testbed* IMS yang terdapat di TELKOM RDC masih menggunakan cara manual, yakni dengan membuka file konfigurasi satu persatu dengan menggunakan text editor, semisal notepad++. Dengan menggunakan teknik seperti ini akan menimbulkan beberapa masalah, diantaranya adalah :

1. Potensi human error yang tinggi, karena banyaknya parameter yang harus diingat dan banyaknya file yang harus diperhatikan.
2. Memerlukan waktu yang lebih untuk mengkonfigurasi semua file konfigurasi tersebut.

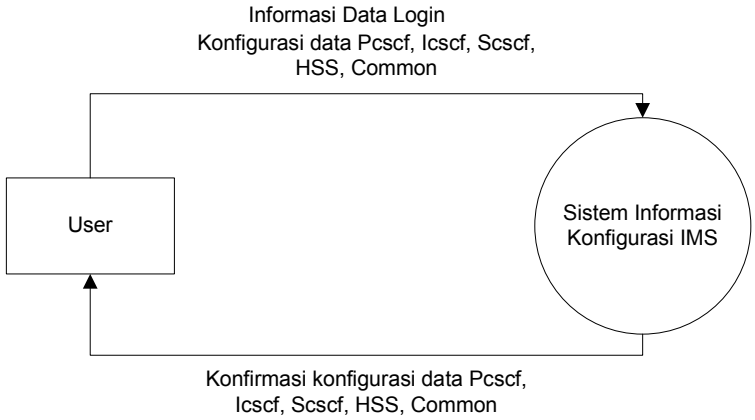
3.1.3 Analisis kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk mengetahui proses informasi yang mengalir melalui perangkat lunak. Untuk menggambarkan proses informasi secara umum digunakan alat bantu, yaitu *Context Diagram* (Diagram Konteks), *Data Flow Diagram* (DFD), kamus data dan spesifikasi proses.

3.1.3.1 Diagram Konteks

Diagram konteks atau disebut juga dengan model sistem fundamental merepresentasikan seluruh elemen sistem sebagai sebuah bubble tunggal dengan data input output yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan.

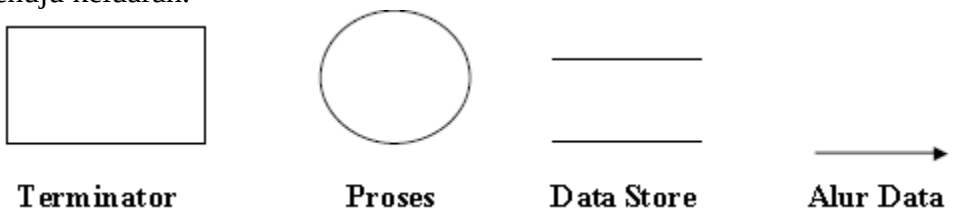
Diagram konteks dari aplikasi yang akan dibangun adalah sebagai berikut :



Gambar 3.1 Diagram konteks

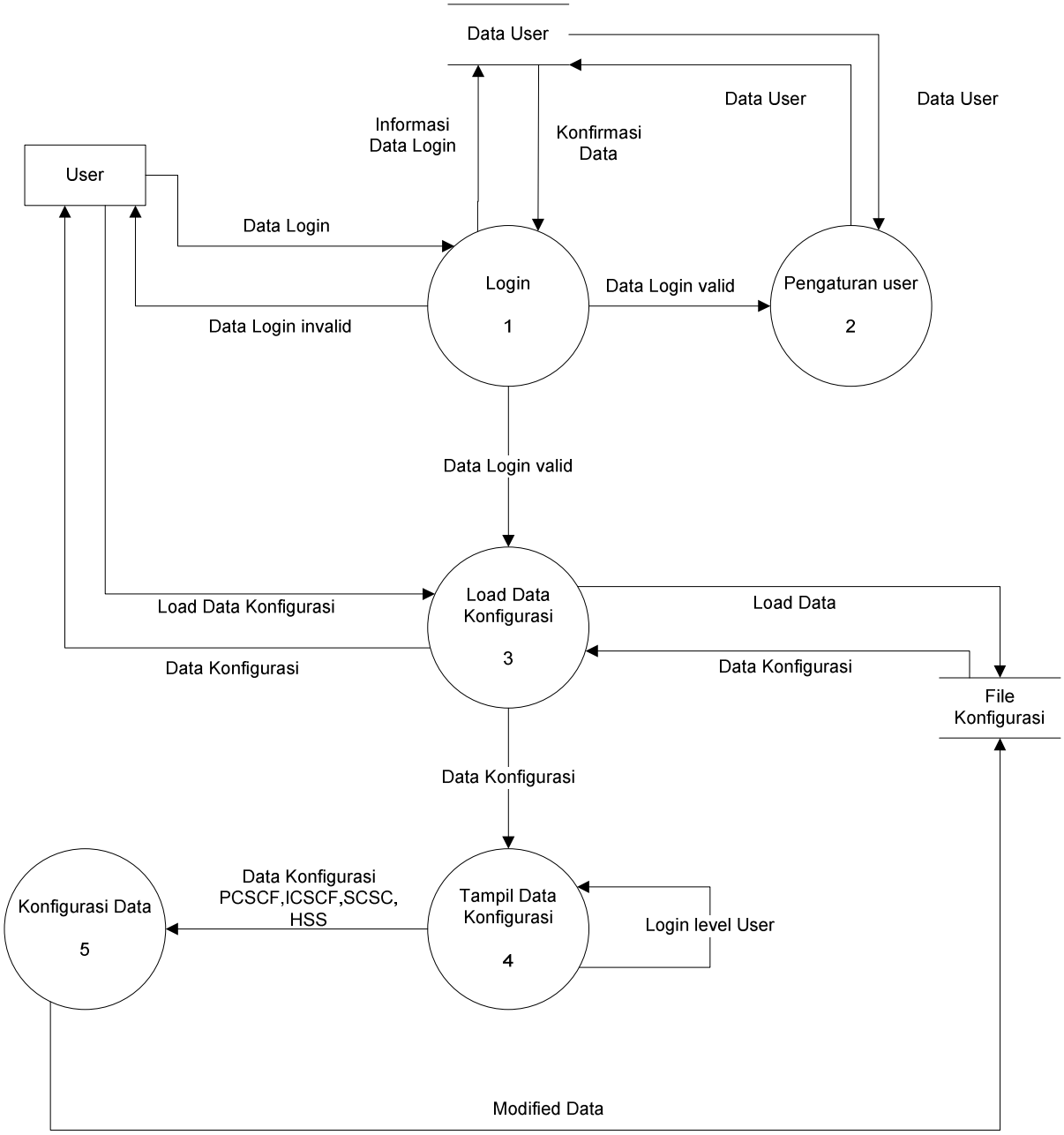
3.1.3.2 Data Flow Diagram

Data Flow Diagram **(**DFD) adalah sebuah teknik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju keluaran.



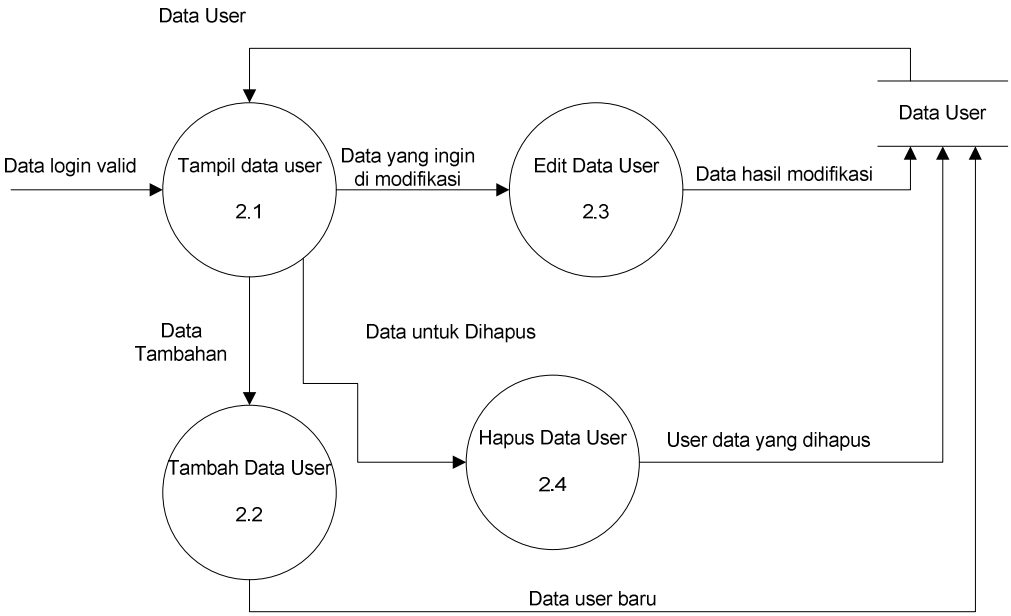
Gambar 3.2 Komponen DFD

3.1.3.2.1 DFD level 0

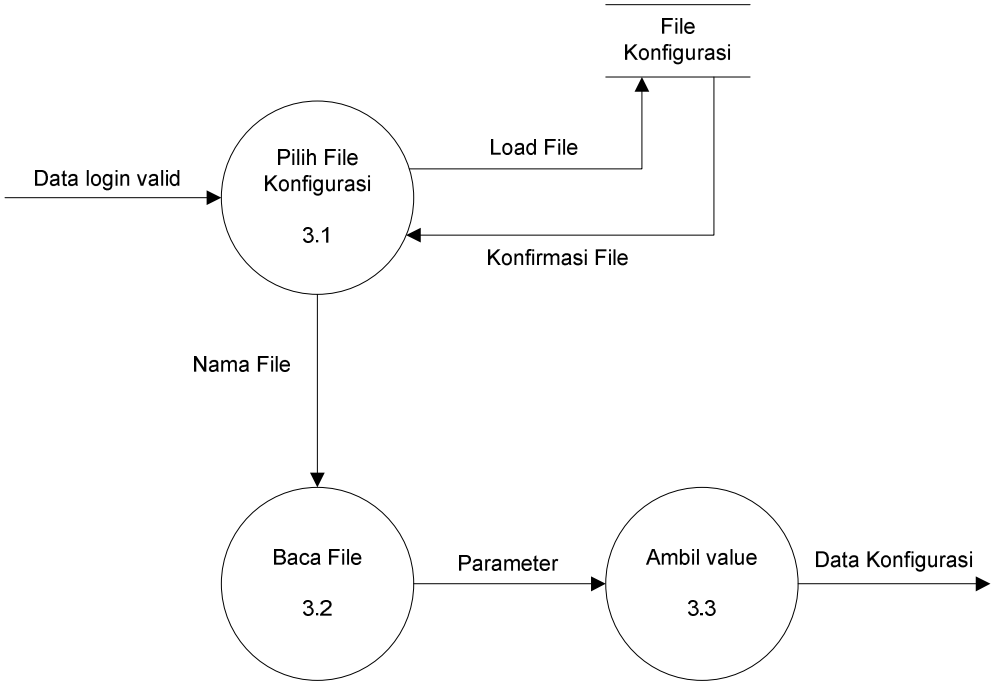


Gambar 3.3 DFD level 0

3.1.3.2.2 DFD level 1



Gambar 3.4 DFD level 1 proses 2.0



Gambar 3.5 DFD level 1 proses 3.0

3.1.3.3 Spesifikasi Proses

No	Proses	Keterangan
1	No.Proses	1.0
	Nama proses	Login
	Source	Super admin, admin, user
	Input	Username dan Password
	Output	Jika Pengisian username dan password valid maka akan dilanjutkan ke proses 2 untuk pengaturan user yang selanjutnya dapat ke proses 3, jika tidak valid maka akan meminta login ulang
	Destination	Super admin, admin, user

	Logika Proses	**Begin**
		{Super admin, admin, user mengisi username dan password, sistem akan melakukan konfirmasi kecocokan pada database }
		If username dan password benar **then**
		{Tampilkan halaman load file konfigurasi *testbed* IMS}
		Else username dan password salah is ulang form login.
		End if
		End

2	No.Proses	2.0
	Nama proses	Pengaturan user
	Source	Super admin dan admin
	Input	Data login valid
	Output	Data user
	Destination	Super admin dan admin
	Logika Proses	**Begin**

		If user = Admin atau Super Admin **then**
		{Tampilkan halaman pengaturan user}
		Else {Tampilkan halaman user biasa}
		End if
		End

3	No.Proses	2.1
	Nama proses	Tampil data user
	Source	Super admin dan admin
	Input	Data Login valid
	Output	Data user
	Destination	Super admin dan admin

	Logika Proses	**Begin**
		{Sistem melakkukan pengecekkan ke database untuk melihat isi dari table user}
		If data user >= 0 **then**
		{Tampilkan data user di tabel user}
		Else {Belum ada data user}
		End if
		End

4	No.Proses	2.2
	Nama proses	Tambah Data User
	Source	Super admin dan admin
	Input	Data tambahan
	Output	Data user baru
	Destination	Super admin dan admin
	Logika Proses	**Begin** {Super Admin atau admin melakkukan penambahan data dengan memasukkan data-data user}

		If data user belum ada **then**
		{Tambahkan data user ke tabel user}
		Else {Data user sudah ada}
		End if
		End

5	No.Proses	2.3
	Nama proses	Edit Data User
	Source	Super admin dan admin
	Input	Data yang ingin dimodifikasi
	Output	Data hasil modifikasi
	Destination	Super admin dan admin

	Logika Proses	**Begin**
		{Super Admin atau admin melakkukan pengeditan data dengan memasukkan data-data baru user }
		If edit berhasil **then**
		{ Edit data user ke tabel user sesuai NIK}
		Else {Edit user gagal }
		End if
		End

6	No.Proses	2.4
	Nama proses	Hapus Data User
	Source	Super admin dan admin
	Input	Data untuk dihapus
	Output	User data yang dihapus
	Destination	Super admin dan admin

	Logika Proses	**Begin**
		{Super Admin atau admin melakkukan penghapusan data dengan menekan delete data user }
		If hapus berhasil **then**

		{ Hapus data user di tabel user sesuai NIK } **Else** {Penghapusan Gagal} **End if** **End**
7	No.Prosess	3.0
	Nama proses	Load data konfigurasi
	Source	Super admin, admin, user
	Input	Data login valid
	Output	Data konfigurasi
	Destination	Super admin, admin, user

	Logika Proses	**Begin**
		{Sistem mencari file konfgurasi di directory}
		If Data konfigurasi = ketemu **then**
		{Ambil data konfiguras}
		Else {Load Data Gagal}
		End if
		End

8	No.Prosess	3.1
	Nama proses	Pilih file konfigurasi
	Source	Super admin, admin, user
	Input	Data login valid
	Output	Nama File
	Destination	Super admin, admin, user

	Logika Proses	**Begin**
		{Super admin, Admin, memilih konfigurasi}
		If pilih konfigurasi **then**
		{Tampil halaman konfigurasi sesuai *testbed* IMS yang dipilih}
		Endif

		End
9	No.Prosess	3.2
	Nama proses	Baca File
	Source	Super admin, admin, user
	Input	Nama file
	Output	Parameter
	Destination	Super admin, admin, user

	Logika Proses	**Begin**
		{Sistem melakukan read dari file konfigurasi}
		If Baca file = berhasil **then**
		{ Sistem melakukan pembacaan file menurut nama file sesuai konfigurasi yang dipilih}
		Else {Baca file Gagal}
		Endif
		End

10	No.Prosess	3.3
	Nama proses	Ambil value
	Source	Super admin, admin, user
	Input	Parameter
	Output	Data Konfigurasi
	Destination	Super admin, admin, user

	Logika Proses	**Begin**
		{Sistem membandingkan parameter di setiap baris untuk mengambil value}
		If Parameter sesuai **then**
		{Ambil nilai}
		Else {Parameter tidak ditemukan}
		Endif
		end

11	No.Proses	4.0
	Nama proses	Tampil data konfigurasi
	Source	Super admin, admin,user
	Input	Data Konfigurasi
	Output	Data konfigurasi untuk dimodifikasi
	Destination	Super admin, admin, user
	Logika Proses	**Begin** {Sistem menampilkan data konfigurasi di tampilan} **End**
12	No.Proses	5.0
	Nama proses	Konfigurasi data
	Source	Super admin dan admin
	Input	Value baru
	Output	Data konfigurasi yang telah diperbaharui
	Destination	Super admin dan admin

	Logika Proses	**Begin**
		{Admin mengubah value parameter pada file konfigurasi}
		If file akan di simpan **then**
		{Simpan data ke file}
		End if
		End

Table 1. Spesifikasi proses

3.1.3.4 Kamus data

Kamus data merupakan katalog (tempat penyimpanan) dari elemen-elemen yang berada dalam suatu sistem. Kamus data mempunyai fungsi yang sama dalam pemodelan sistem dan juga berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

No	Kamus	Keterangan
1	Nama Aliran data	Data Login
	Where	- Dari proses 1 ke proses 2 untuk pengaturan user. - Dari proses 1 ke proses 3 untuk load data konfigurasi
	Deskripsi	Berisi identitas Super Admin, Admin dan User
	Struktur data	
	Id_user Password	9[A..Z|a..z|0..9] 21[A..Z|a..z|0..9]
2	Nama Aliran data	Data Konfigurasi P-CSCF
	Where	- Dari proses 4 ke proses 5 untuk pengeditan data P-CSCF - Dari proses 5 ke penyimpanan file konfigurasi untuk menyimpan value baru
	Deskripsi	Berisi data konfigurasi P-CSCF
	Struktur data	Bervariasi sesuai

		parameter yang ada di PCSCF
	Memlog,debug,children. SIP_warning,log_stderror, fork,check_via,dns,rev_dns, RTP Porxy, TLS . Nama. IP I-CSCF, IP TLS, IP RTP Proxy. SIP Port, TLS Port. RTP Proxy Port	1[|0..9] 3[yes/no] 23[a..z] 16[|0..9] 4[|0..9] 6[|0..9]
3	Nama Aliran data	Data Konfigurasi I-CSCF
	Where	- Dari proses 4 ke proses 5 untuk pengeditan data I-CSCF - Dari proses 5 ke penyimpanan file konfigurasi untuk menyimpan value baru
	Deskripsi	Berisi data konfigurasi I-CSCF
	Struktur data	Bervariasi sesuai parameter yang ada di I-CSCF
	Memlog,debug,children. SIP_warning,log_stderror, fork,check_via,dns,rev_dns. Nama. IP SIP Port, Diameter Port.	1[|0..9] 3[yes/no] 23[a..z] 16[|0..9] 4[|0..9]
4	Nama Aliran data	Data Konfigurasi S-CSCF

	Where	- Dari proses 4 ke proses 5 untuk pengeditan data S-CSCF - Dari proses 5 ke penyimpanan file konfigurasi untuk menyimpan value baru
	Deskripsi	Berisi data konfigurasi S-CSCF
	Struktur data	Bervariasi sesuai parameter yang ada di S-CSCF
	Memlog,debug,children. SIP_warning,log_stderror, fork,check_via,dns,rev_dns. Nama. IP SIP Port, Diameter Port, Reg Defaul Expires. Reg Default Algorithm.	1[|0..9] 3[yes/no] 23[a..z] 16[|0..9] 4[|0..9] 23[|a..z,A..Z,0..9]
5	Nama Aliran data	Data Konfigurasi HSS
	Where	- Dari proses 4 ke proses 5 untuk pengeditan data HSS - Dari proses 5 ke penyimpanan file konfigurasi untuk menyimpan value baru
	Deskripsi	Berisi data konfigurasi HSS
	Struktur data	Bervariasi sesuai parameter yang ada di

		HSS
	Nama. IP. Diameter Port, Subcription expiry time, Dialogs expiration time. Web server Port.	23[a..z] 16[|0..9] 4[|0...9] 2[|0..9]

Tabel 2. Kamus data

3.1.4 Analisis Kebutuhan non – Fungsional

Tahap ini meliputi analisis terhadap elemen-elemen apa saja yang dibutuhkan untuk membangun aplikasi untuk melihat dan mengubah file konfigurasi *testbed* IMS di TELKOM RDC. Kebutuhan non-fungisonal terbagi menjadi tiga yaitu kebutuhan perangkat keras, kebutuhan perangkat lunak dan kebutuhan pengguna.

3.1.4.1 Analisis Perangkat Lunak

Perangkat lunak merupakan hal yang penting dalam mendukung kinerja sebuah sistem. Sistem yang digunakan di TELKOM RDC minimal mempunyai perangkat lunak sebagai berikut :

a) Microsoft Windows XP Profesional Service Pack 1 .
b) Microsoft Office.
c) Wamp server untuk menjalankan program konfigurasi IMS.

Dari hasil observasi, sistem yang ada pada TELKOM RDC sudah dilengkapi dengan perangkat lunak yang dipersyaratkan tersebut.

3.1.4.2 Analisis Perangkat Keras

Perangkat keras atau *hardware* merupakan salah satu hal yang sangat penting, karena tanpa *hardware* yang memenuhi syarat, aplikasi yang akan dibangun tidak akan bisa berjalan. Berikut spesifikasi perangkat keras yang ada pada TELKOM RDC yaitu :

a) Monitor 14“
b) Processor Intel Pentium IV 2,6 Ghz'
c) Hardisk Drive 80 Gb
d) Memory/RAM 1 Gb
e) VGA Card 256 Mb
f) CD ROM drive
g) Keyboard
h) Mouse

Sedangkan minimal *hardware* untuk melakukan instalasi dan menggunakan PHP dan MySQL dengan normal, persyaratannya sebagai berikut :

a) Prosesor Pentium 166MHz atau yang lebih tinggi.
b) System operasi Windows 98, Windows 98 Se, Windows ME, Windows 2000, Windows 2003, dan Windows XP
c) Monitor
d) Hardisk minimal size yang kosong sebesar 500 Mb
e) Memory/RAM 256 Mb
f) VGA atau resolusi monitor 1024x768
g) CD ROM Drive
h) Keyboard
i) Mouse

3.1.4.3 Analisis Pengguna

Untuk studi kasus di TELKOM RD, karyawan harus dimungkinkan untuk melakukan eksplorasi *testbed* IMS sesuai dengan kebutuhanya masing – masing, ada yang hanya bertujuan untuk belajar mengenai *testbed* IMS, melihat file konfigurasi *testbed* IMS dan lain sebagainya. Untuk itu, perlu adanya klasifikasi pengguna berdasarkan kebutuhannya. Setelah melakukan diskusi dengan pembimbing yang ada di lapangan, pengguna aplikasi akan dibedakan ke dalam tiga level, yakni :

1. *Super admin* berperan untuk menambah, menghapus dan mengontrol aktivitas admin dan memiliki akses penuh dimana dapat mengubah file konfigurasi, menambah dan menghapus *admin* dan *User* Lainnya.
2. *Admin* adalah pengguna yang dapat mengubah file konfigurasi dan menambah dan menghapus *user*.
3. *User* adalah pengguna yang hanya dapat melihat file konfigurasi.

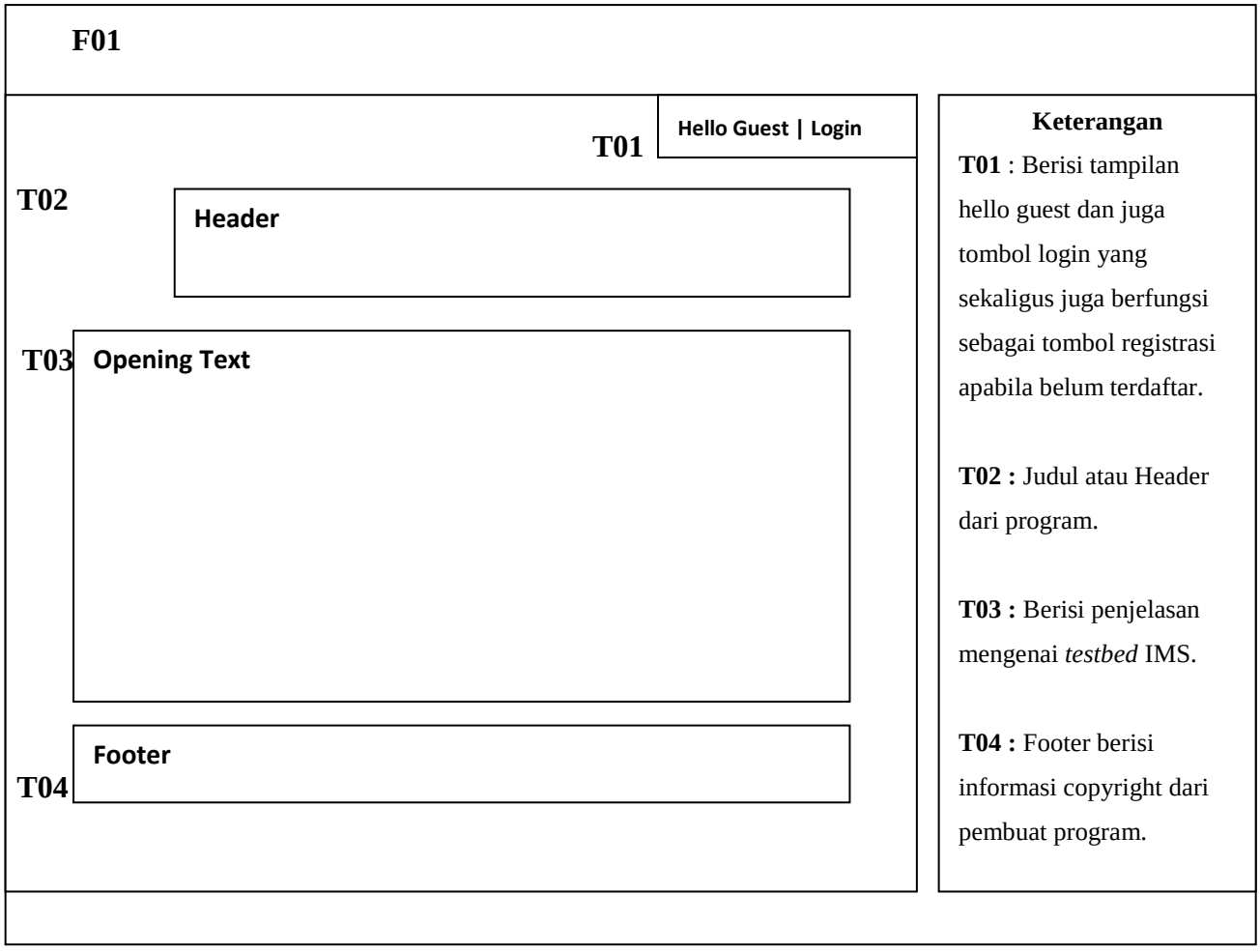
3.2 Perancangan sistem

Perancangan sistem, bertujuan untuk menerapkan solusi pemecahan masalah yang telah diajukan pada analisis sistem.

3.2.1 Perancangan Antar Muka

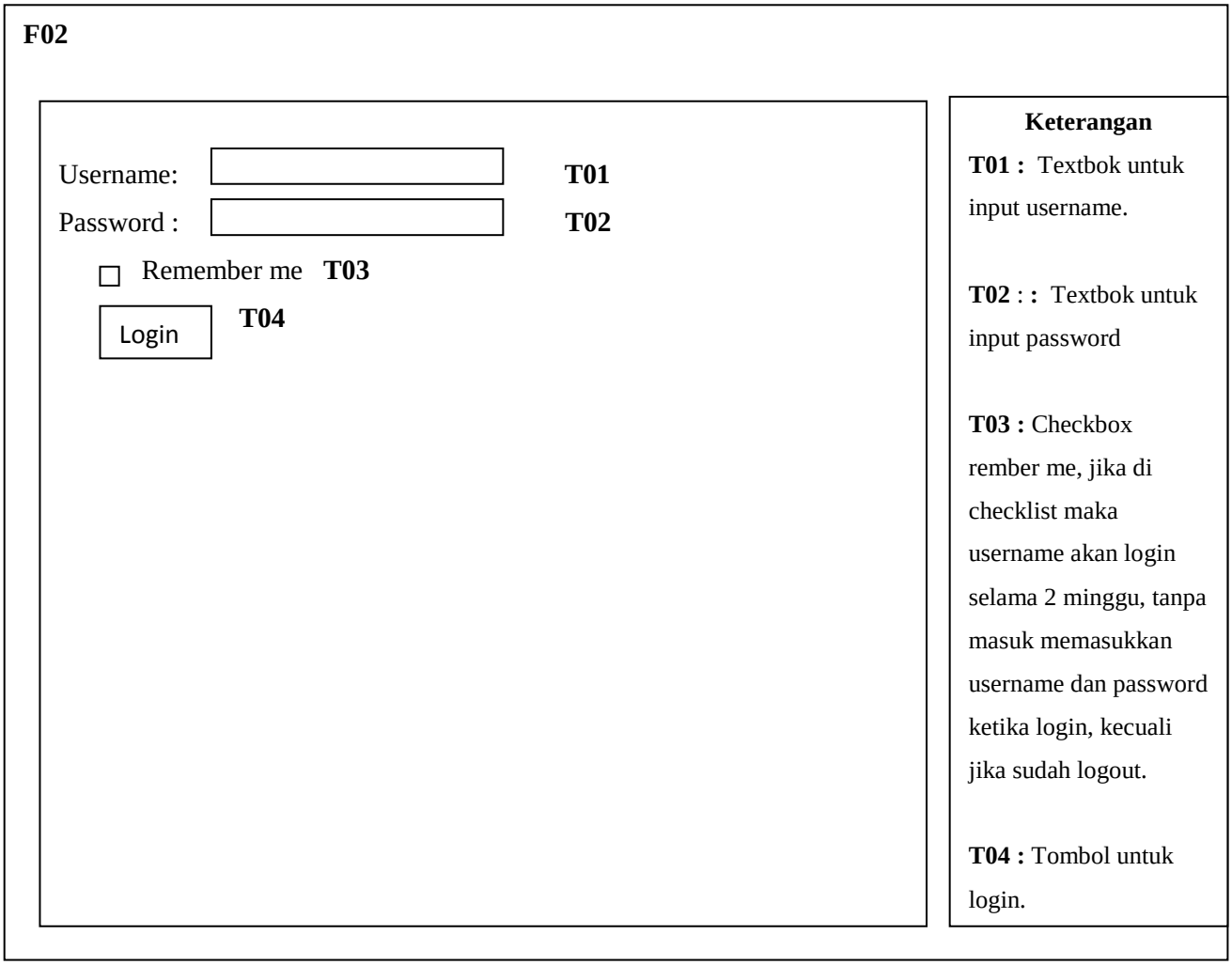
Perancangan antarmuka bertujuan untuk memberikan gambaran mengenai apa yang akan dibuat dalam pembuatan interface aplikasi.

1. Perancangan Tampilan Utama



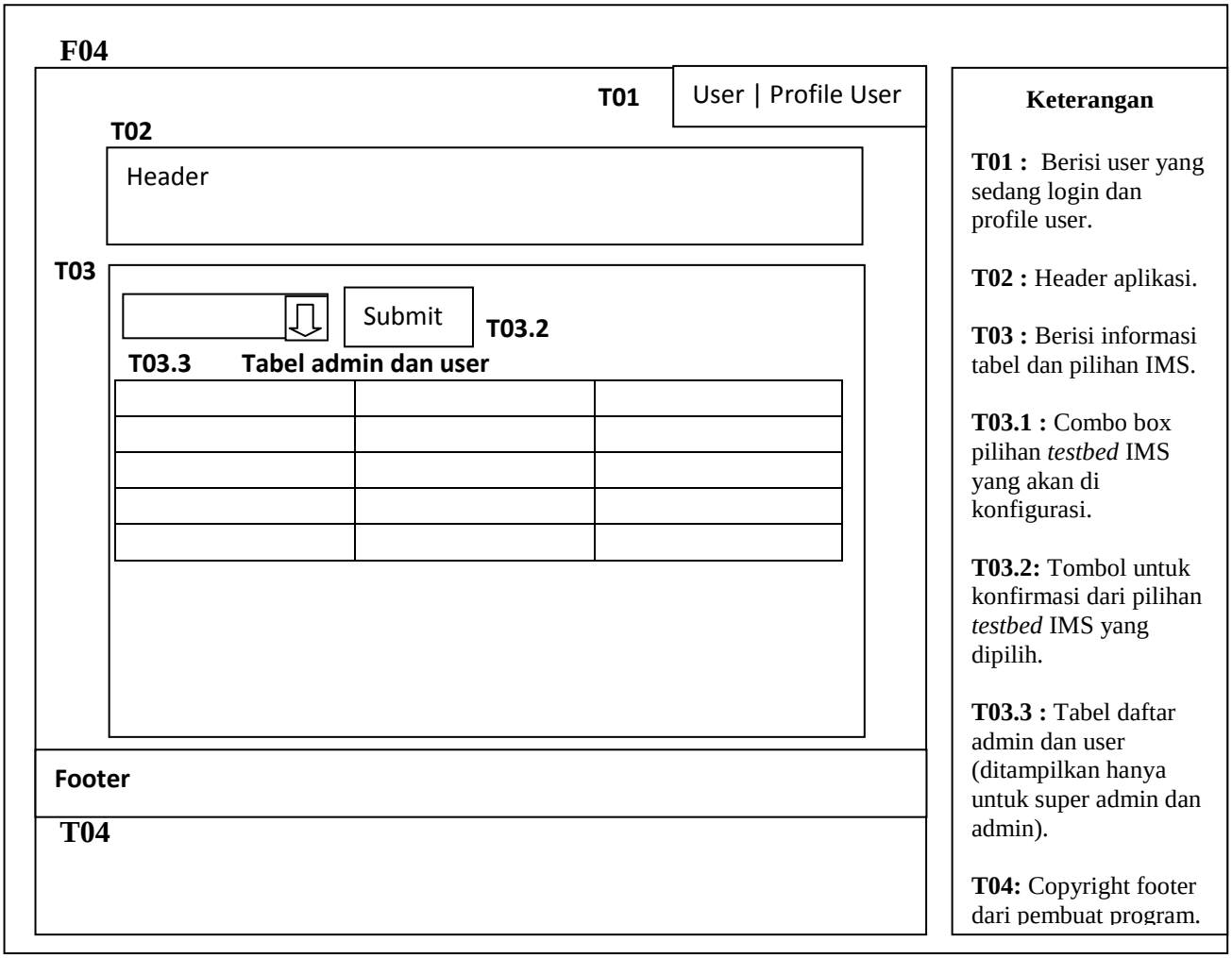
Gambar 3.6 Rancangan Tampilan utama

2. Perancangan Tampilan Login



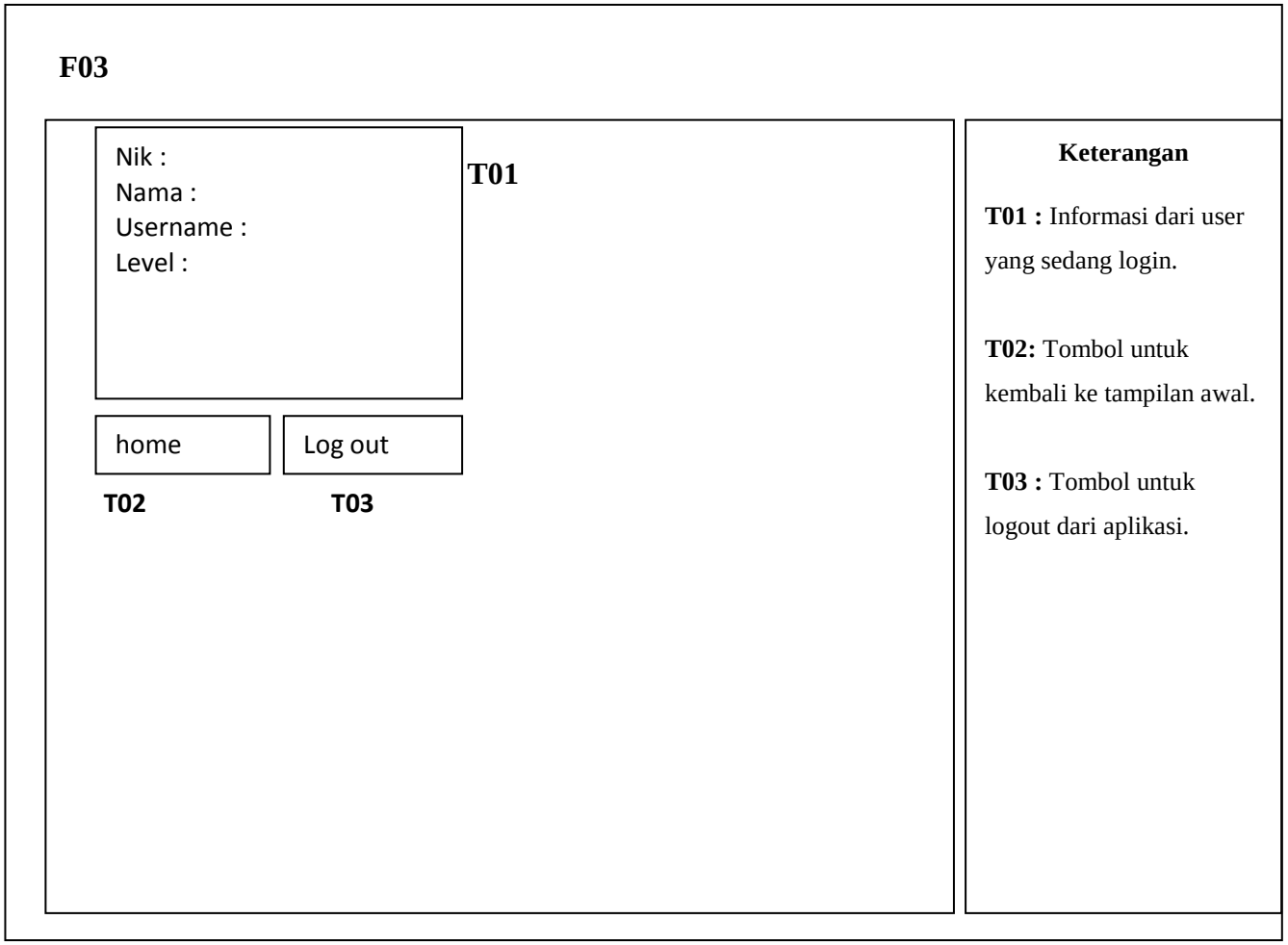
Gambar 3.7 Rancangan Tampilan login

3. Perancangan Tampilan Login Super admin,Admin, dan User



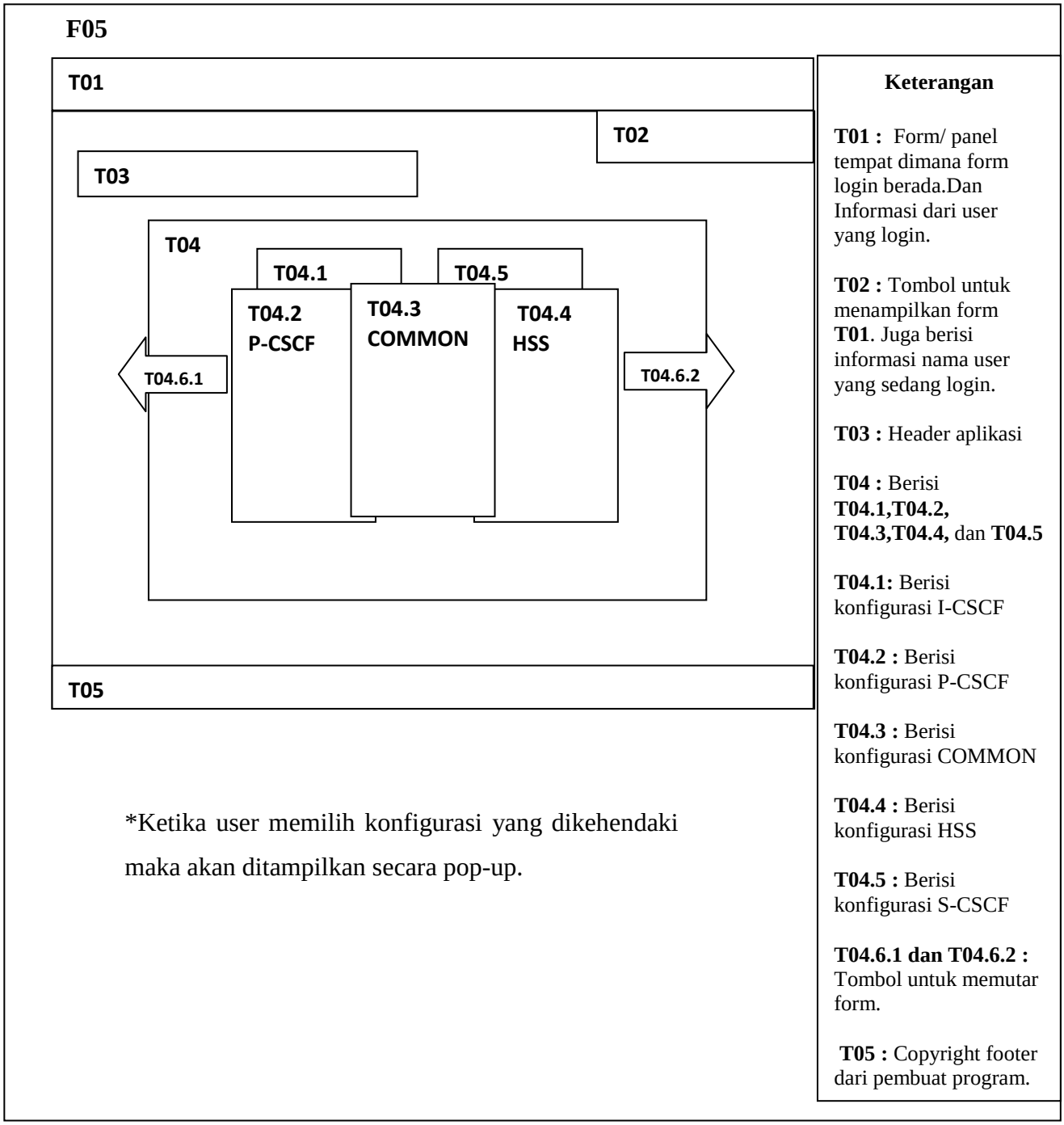
Gambar 3.8 Rancangan Tampilan login super admin, admin, user.

4. Perancangan Tampilan Profil user



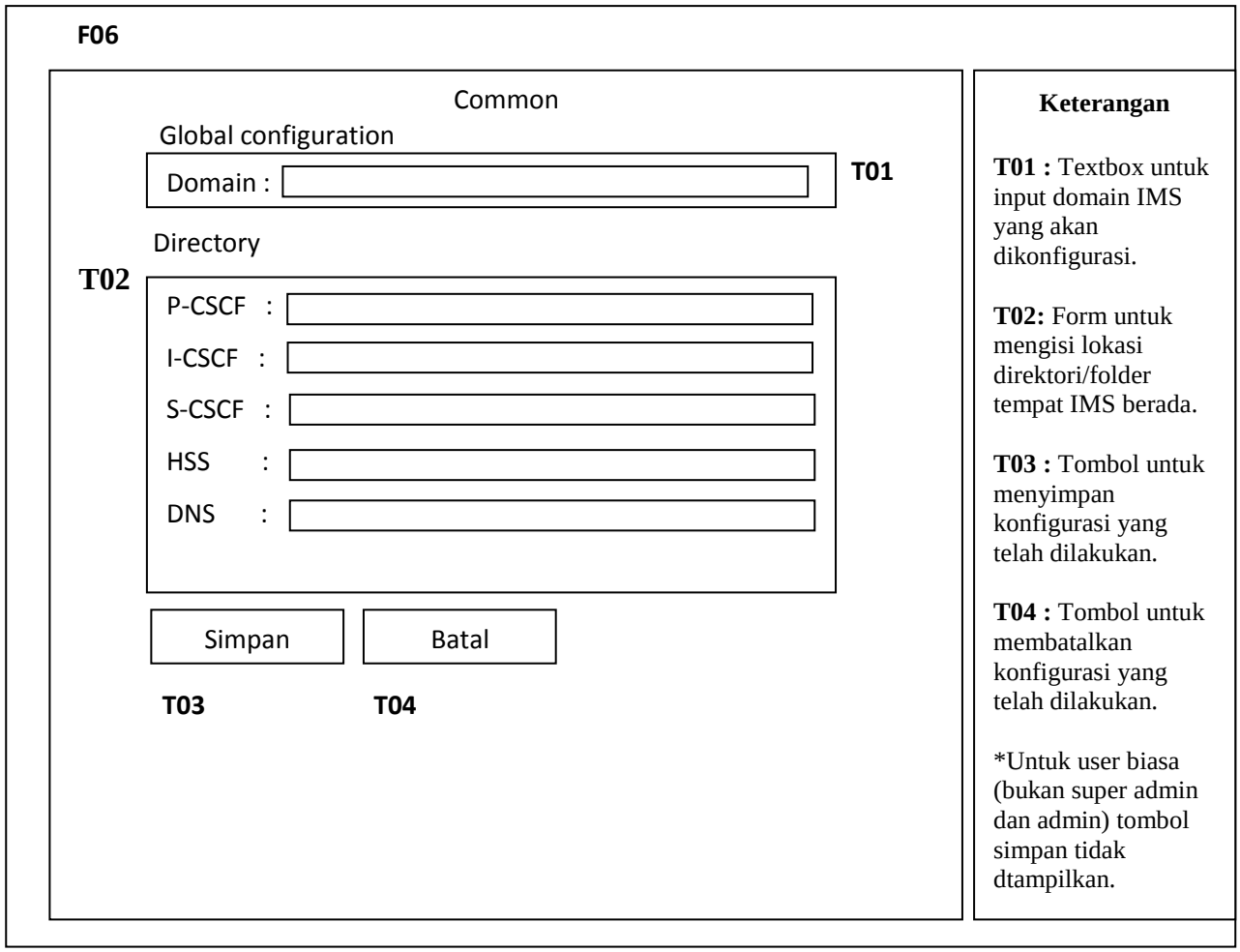
Gambar 3.9 Rancangan tampilan profil user

5. Perancangan Tampilan Konfigurasi IMS



Gambar 3.10 Rancangan tampilan konfigurasi IMS

6. Perancangan Tampilan COMMON configuration



Gambar 3.11 Rancangan Tampilan Common configuration

Pada form common configuration di bagi dua kelompok konfigurasi:

1. Global configuration

Pada global konfiguration terdapat domain yang menunjukan domain *testbed* IMS yang akan dikonfigurasi.

a) *Value* dari Domain diambil (load) dari parameter *realm* yang terdapat di file konfigurasi scscf.xml

b) Jika *value* Domain diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok nama domain akan secara otomatis diubah oleh sistem. Anggota kelompok nama domain ini telah di definisikan di

bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

2. Directory

Directory menunjukan lokasi *testbed* IMS berada. Pada konfigurasi *testbed* IMS ada beberapa directory file lokasi *testbed* IMS, diantaranya adalah :

a) P-CSCF

i. *Value* dari P-CSCF diambil (load) dari parameter *pcscf* yang terdapat di file konfigurasi ims1.cfg

ii. Jika *value* P-CSCF diubah dan kemudian disimpan, maka *value pcscf* yang terdapat pada ims1.cfg akan secara otomatis diubah oleh sistem. Parameter *pcscf* pada file ims1.cfg tidak dikelompokkan karena hanya merupakan parameter pendukung aplikasi.

b) I-CSCF

i. *Value* dari I-CSCF diambil (load) dari parameter *icscf* yang terdapat di file konfigurasi ims1.cfg

ii. Jika *value* I-CSCF diubah dan kemudian disimpan, maka *value icscf* yang terdapat pada ims1.cfg akan secara otomatis diubah oleh sistem. Parameter *icscf* pada file ims1.cfg tidak dikelompokkan karena hanya merupakan parameter pendukung aplikasi.

c) S-CSCF

i. *Value* dari S-CSCF diambil (load) dari parameter *scscf* yang terdapat di file konfigurasi ims1.cfg

ii. Jika *value* S-CSCF diubah dan kemudian disimpan, maka *value scscf* yang terdapat pada ims1.cfg akan secara otomatis diubah oleh sistem. Parameter *scscf* pada file ims1.cfg tidak dikelompokkan karena hanya merupakan parameter pendukung aplikasi.

d) HSS

i. *Value* dari HSS diambil (load) dari parameter *hss* yang terdapat di file konfigurasi ims1.cfg

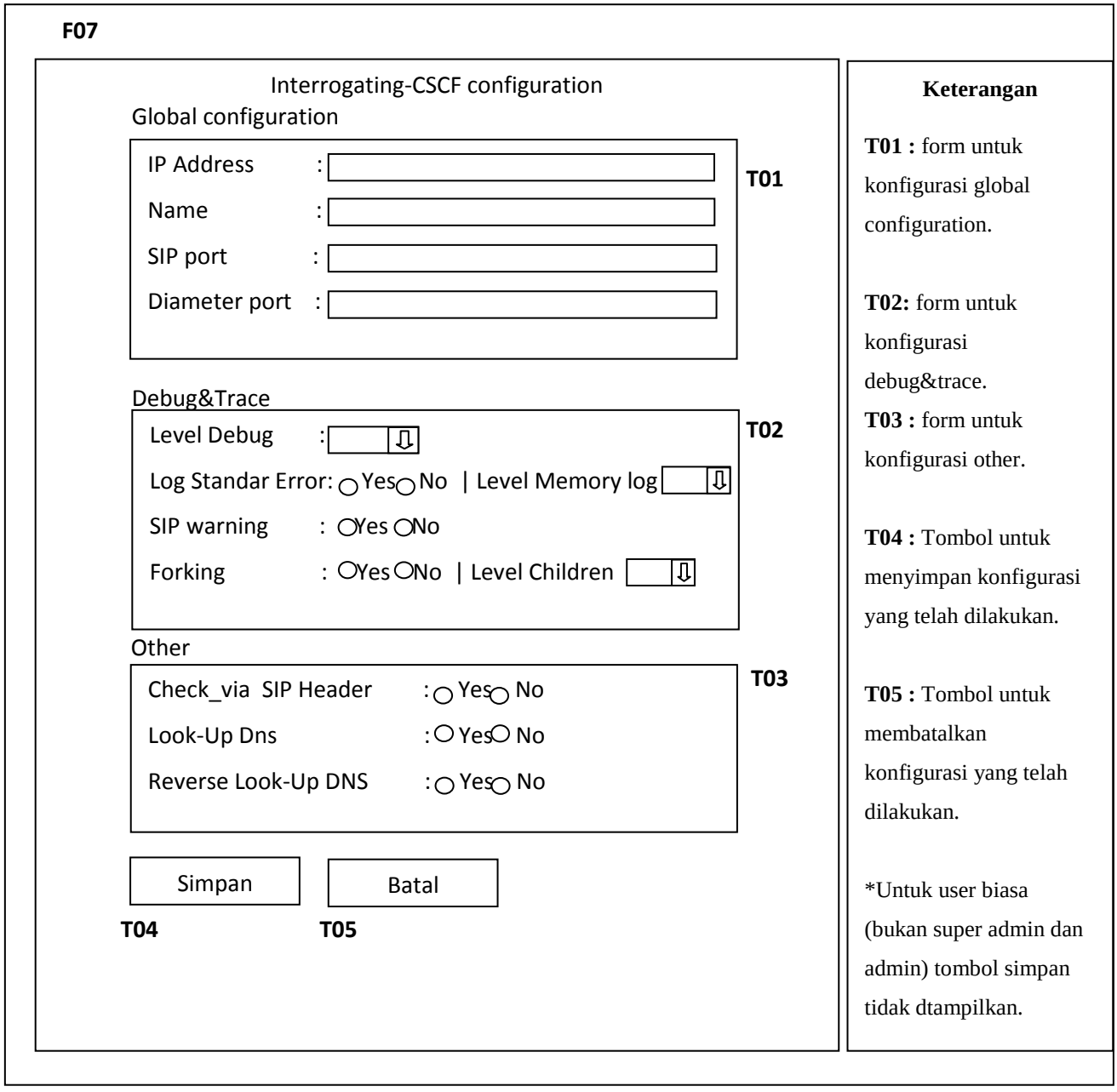
ii. Jika *value* HSS diubah dan kemudian disimpan, maka *value hss* yang terdapat pada ims1.cfg akan secara otomatis diubah oleh sistem. Parameter *hss* pada file ims1.cfg tidak dikelompokkan karena hanya merupakan parameter pendukung aplikasi.

e) DNS

i. *Value* dari DNS diambil (load) dari parameter *named* yang terdapat di file konfigurasi ims1.cfg

ii. Jika *value* DNS diubah dan kemudian disimpan, maka *value dns* yang terdapat pada ims1.cfg akan secara otomatis diubah oleh sistem. Parameter *named* pada file ims1.cfg tidak dikelompokkan karena hanya merupakan parameter pendukung aplikasi.

7. Perancangan Tampilan I-CSCF configuration



Gambar 3.12 Rancangan Tampilan I-CSCF configuration

Pada form I-CSCF configuration di bagi 3 kelompok konfigurasi:

1. Global confguration

a) IP Address

i. *Value* dari IP Address diambil (load) dari parameter *listen* yang terdapat di file konfigurasi icscf.cfg

ii. Jika value IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok IP Address I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok IP Address I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

***b)* Name**

i. *Value* dari Name diambil (load) dari parameter *FQDN* yang terdapat di file konfigurasi icscf.xml

ii. Jika *value* Name diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Nama I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Nama I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan

c) SIP Port

i. *Value* dari SIP Port I-CSCF diambil (load) dari parameter *port* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika SIP Port I-CSCF diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok SIP Port I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok SIP Port I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan

d) Diameter port

i. *Value* dari Diameter Port I-CSCF diambil (load) dari parameter *acceptor port* yang terdapat di file konfigurasi icscf.xml.

ii. Jika Diameter Port I-CSCF diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Diameter Port I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Diameter Port I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan

2. Debug&Trace

a) Level Debug

i. *Value* dari Level Debug diambil (load) dari parameter *debug* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika Level Debug diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level debug di I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level debug di I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Log Standar Error

i. *Value* dari Log Standar Error diambil (load) dari parameter *log_stderror* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* Log Standar Error diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Log Standar Error I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Log Standar Error I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Level Memory Log

Level Memory Log bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi Log Satandar Error.

i. *Value* dari Level Memory Log diambil (load) dari parameter *memlog* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* Level Memory Log diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Memory Log di I-CSCF

akan secara otomatis diubah oleh sistem. Anggota kelompok Level Memory Log di I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

d) SIP Warning

i. *Value* dari SIP Warning diambil (load) dari parameter *sip_warning* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* SIP Warning diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Sip Warning I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Sip Warning I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

e) Forking

i. *Value* dari Forking diambil (load) dari parameter *fork* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* Forking diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Forking I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Forking I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

f) Level Children

Level Children bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi forking.

i. *Value* dari Level Children diambil (load) dari parameter *children* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* Level Children diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Children I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level Children I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

3. Other

a) Check via SIP Header

i. *Value* dari Check via SIP Header diambil (load) dari parameter *check_via* yang terdapat di file konfigurasi icscf.cfg.

ii. Jika *value* Check via SIP Header diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Check via SIP Header I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Check via SIP Header I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Look-Up DNS

i. *Value* dari Look-Up DNS diambil (load) dari parameter *dns* yang terdapat di file konfigurasi icscf.cfg

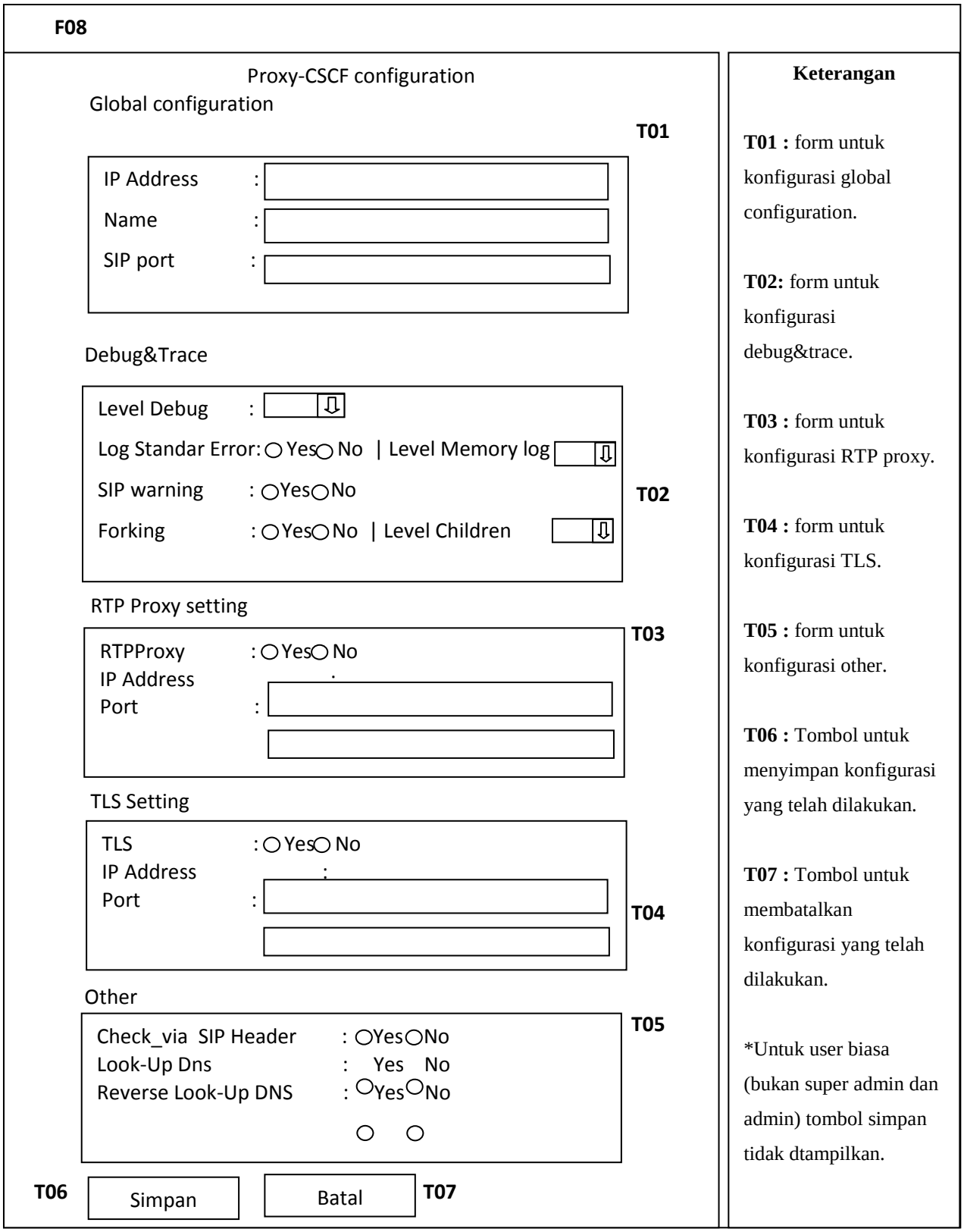
ii. Jika *value* Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Look-Up DNS I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Look-Up DNS I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Reversed Look-Up DNS

i. *Value* dari Reversed Look-Up DNS diambil (load) dari parameter *rev_dns* yang terdapat di file konfigurasi icscf.cfg

ii. Jika *value* Reversed Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Reversed Look-Up DNS I-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Reversed Look-Up DNS I-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

8. Perancangan Tampilan P-CSCF configuration



Gambar 3.13 Rancangan Tampilan P-CSCF configuration

Pada form P-CSCF configuration di bagi 5 kelompok konfigurasi:

1. Global confguration

a) IP Address

i. *Value* dari IP Address diambil (load) dari parameter *listen* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika value IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok IP Address P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok IP Address P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) **Name**

i. *Value* dari name diambil (load) dari parameter *alias* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Name diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Nama P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Nama P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan

c) SIP port

i. *Value* dari SIP Port diambil (load) dari parameter *port* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* SIP Port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok SIP Port P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok SIP Port P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan

2. Debug&Trace

a) Level Debug

i. *Value* dari Level Debug diambil (load) dari parameter *debug* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Level Debug diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level debug di P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level debug di P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Log Standar Error

i. *Value* dari Log Standar Error diambil (load) dari parameter *log_stderror* yang terdapat di file konfigurasi pcscf.cfg.

ii. Jika *value* Log Standar Error diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Log Standar Error P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Log Standar Error P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Level Memory Log

Level Memory Log bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi Log Standar Error.

i. *Value* dari Level Memory Log diambil (load) dari parameter *memlog* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Level Memory Log diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Memory Log di P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level Memory Log di P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

d) SIP Warning

i. *Value* dari SIP Warning diambil (load) dari parameter *sip_warning* yang terdapat di file konfigurasi pcscf.cfg.

ii. Jika *value* SIP Warning diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Sip Warning P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Log Sip Warning P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

e) Forking

i. *Value* dari forking diambil (load) dari parameter *fork* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* forking diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok forking di P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok forking di P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

f) Level Children

Level Children bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi forking.

i. *Value* dari level children diambil (load) dari parameter *children* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* level children diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Children di P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level Children P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

3. RTP Proxy Setting

a) RTP proxy

i. *Value* dari RTP proxy diambil (load) dari parameter *rtpproxy_enable* yang terdapat di file konfigurasi pcscf.cfg.

ii. Jika *value* RTP proxy diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok RTP proxy akan secara otomatis diubah oleh sistem. Anggota kelompok RTP proxy ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) IP Address

IP Address dari RTP Proxy bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi RTP Proxy.

i. *Value* dari IP Address diambil (load) dari parameter *rtpproxy_socket* yang terdapat di file konfigurasi pcscf.cfg, tepatnya di bagian yang menunjukkan IP Address.

ii. Jika *value* IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok RTP proxy akan secara otomatis diubah oleh sistem. Anggota kelompok RTP proxy ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Port

Port dari RTP Proxy bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi RTP Proxy.

i. *Value* dari Port diambil (load) dari parameter *rtpproxy_socket* yang terdapat di file konfigurasi pcscf.cfg, tepatnya pada bagian yang menunjukkan port.

ii. Jika *value* Port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok RTP proxy akan secara otomatis diubah oleh sistem. Anggota kelompok RTP proxy ini telah di definisikan di

bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

4. TLS Setting

a) TLS

i. *Value* dari TLS diambil (load) dari parameter *use_tls* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* TLS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok TLS akan secara otomatis diubah oleh sistem. Anggota kelompok TLS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) IP Address

IP Address dari TLS bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi TLS.

i. *Value* dari IP Address diambil (load) dari parameter *listen_tls* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok TLS akan secara otomatis diubah oleh sistem. Anggota kelompok TLS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Port

Port dari TLS bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi TLS

i. *Value* dari Port diambil (load) dari parameter *tls_port_no* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok TLS akan secara otomatis diubah oleh sistem. Anggota kelompok TLS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

5. Other

a) Check via SIP Header

i. *Value* dari Check via SIP Header diambil (load) dari parameter *check_via* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Check via SIP Header diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok check Via SIP Header P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok check Via SIP Header P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Look-Up DNS

i. *Value* dari Look-Up DNS diambil (load) dari parameter *DNS* yang terdapat di file konfigurasi pcscf.cfg

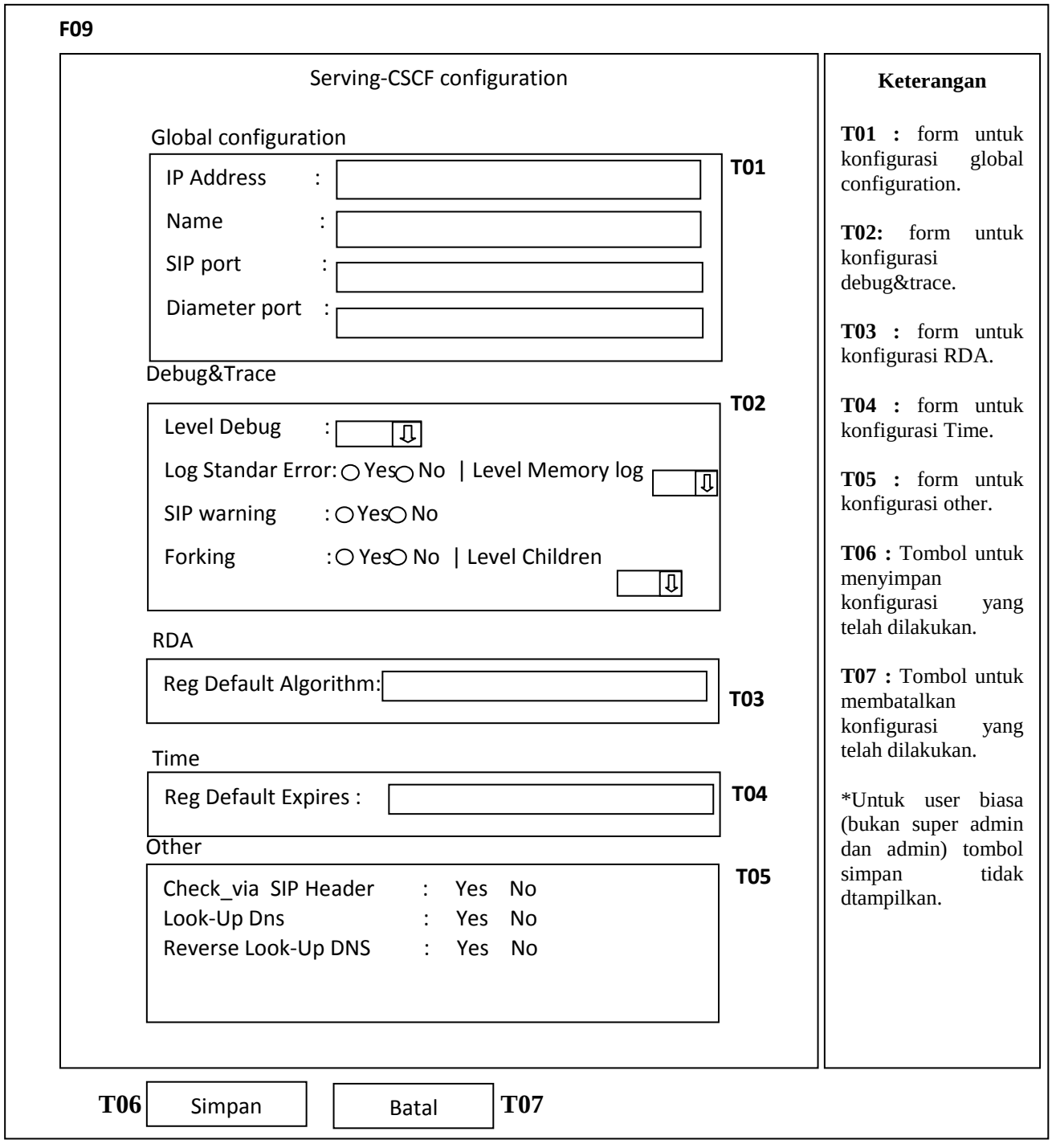
ii. Jika *value* Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Look-Up DNS P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Look-Up DNS P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Reversed Look-Up DNS

i. *Value* dari Reversed Look-Up DNS diambil (load) dari parameter *rev_dns* yang terdapat di file konfigurasi pcscf.cfg

ii. Jika *value* Reversed Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Reversed Look-Up DNS P-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Reversed Look-Up DNS P-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

9. **Perancangan Tampilan S-CSCF configuration**



Gambar 3.14 Rancangan Tampilan S-CSCF configuration

Pada form S-CSCF configuration di bagi 5 kelompok konfigurasi:

1. Global confguration

a) IP Address

i. *Value* dari IP Address diambil (load) dari parameter *listen* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok IP Address S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok IP Address S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Name

i. *Value* dari name diambil (load) dari parameter *FQDN* yang terdapat di file konfigurasi scscf.xml

ii. Jika *value* name diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Nama S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Nama S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) SIP port

i. *Value* dari SIP Port diambil (load) dari parameter *port* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* SIP Port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok SIP Port S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok SIP Port S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

2. Debug&Trace

a) Level Debug

i. *Value* dari Level debug diambil (load) dari parameter *debug* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* Level debug diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Debug di S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level Debug di S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Log Standar Error

i. *Value* dari Log Standar Error diambil (load) dari parameter *log_stderror* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* Log Standar Error diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Log Standar Error S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Log Standar Error S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Level Memory Log

Level Memory Log bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi Log Standar Error.

i. *Value* dari Level Memory Log diambil (load) dari parameter *memlog* yang terdapat di file konfigurasi scscf.cfg

ii. *Jika value* Level Memory Log diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level Memory Log di S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Level Memory Log di S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

d) SIP Warning

i. *Value* dari SIP Warning diambil (load) dari parameter *sip_warning* yang terdapat di file konfigurasi scscf.cfg

ii. *Jika value* SIP Warning diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok SIP Warning S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok SIP Warning S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

e) Forking

i. *Value* dari forking diambil (load) dari parameter *fork* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* forking diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok forking di S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok forking di S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

f) Level Children

Level Children bisa dikonfigurasi oleh user apabila user memilih pilihan yes pada konfigurasi forking.

i. *Value* dari level children diambil (load) dari parameter *children* yang terdapat di file konfigurasi scscf.cfg.

ii. Jika *value* level children diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Level children S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok f Level children di S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

3. RDA

a) Registration Default Algorithm

i. *Value* dari Registration Default Algorithm diambil (load) dari parameter *Registration_default_algorithm dengan pola "modparam("scscf"," registration_default_ algorithm",xxxx)"* yang terdapat di file konfigurasi scscf.cfg. Value yang menjadi nilai default adalah value dari baris yang tidak mempunyai tanda "#" di depannya.

ii. Jika *value* Registration Default Algorithm diubah dan kemudian disimpan, maka mekanisme penyimpananya adalah sebagai berikut:

a) Value yang dipilih akan disimpan tanpa tanda "#" didepannya.

b) Value yang tidak dipilih akan disimpan dengan tanda "#" di depannya.

4. Time

a) Registration Default Expires

i. *Value* dari Registration Default Expires diambil (load) dari parameter *Registration_default_expires dengan pola "modparam("scscf"," registration_default_expires",xxxx)"* yang terdapat di file konfigurasi scscf.cfg.

ii. Jika *value* Registration Default Expires diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Default Time Expiraation of Registration akan secara otomatis diubah oleh sistem. Anggota kelompok Default Time Expiraation of Registration ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

5. Other

a) Check via SIP Header

i. *Value* dari Check via SIP Header diambil (load) dari parameter *check_via* yang terdapat di file konfigurasi scscf.cfg.

ii. Jika *value* Check via SIP Header diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Check via SIP Header S-

CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Check via SIP Header S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Look-Up DNS

i. *Value* dari Look-Up DNS diambil (load) dari parameter *dns* yang terdapat di file konfigurasi scscf.cfg

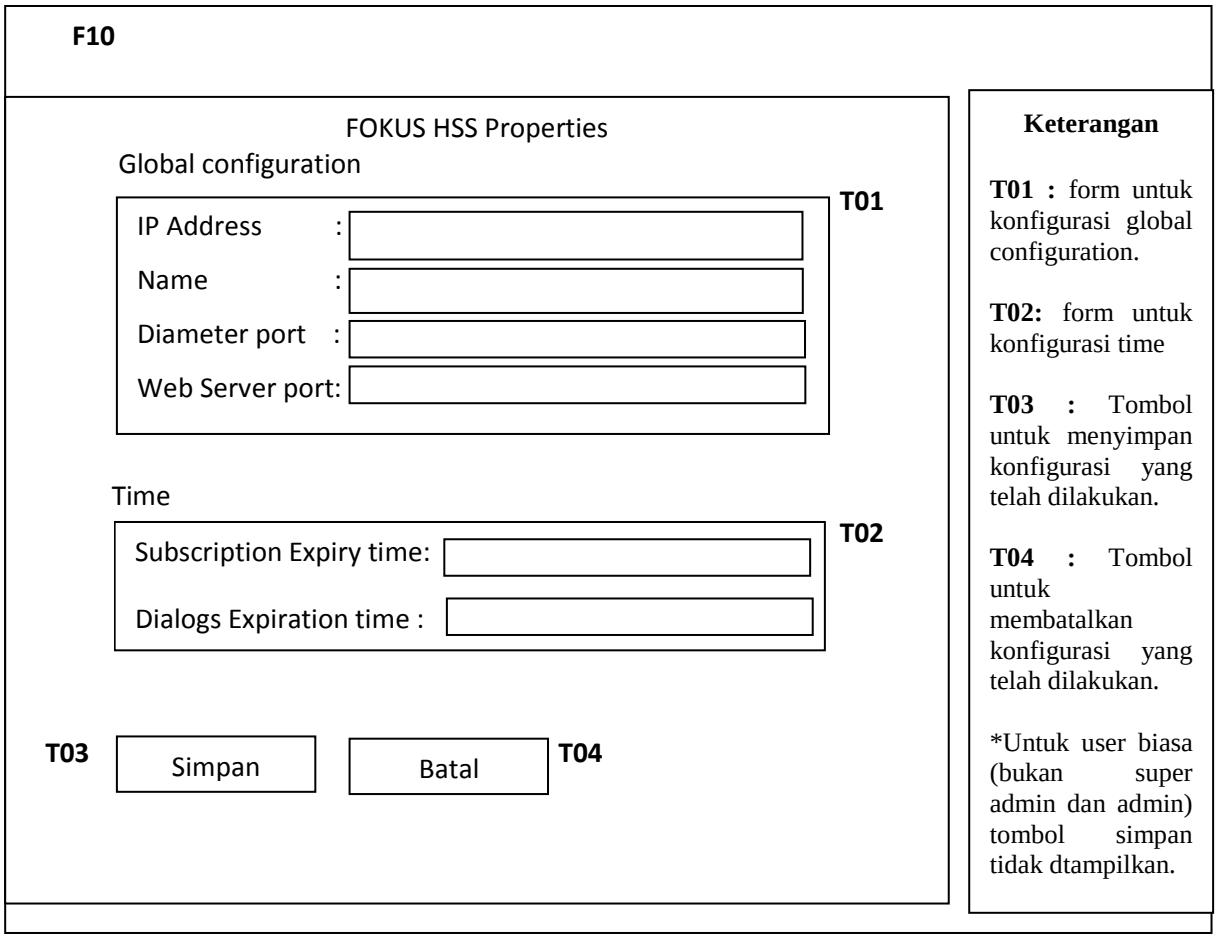
ii. Jika *value* Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Look-Up DNS S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Look-Up DNS S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Reversed Look-Up DNS

i. *Value* dari Reversed Look-Up DNS diambil (load) dari parameter *rev_dns* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* Reversed Look-Up DNS diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Reversed Look-Up DNS S-CSCF akan secara otomatis diubah oleh sistem. Anggota kelompok Reversed Look-Up DNS S-CSCF ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

10. Perancangan Tampilan HSS configuration



Gambar 3.15 Rancangan Tampilan HSS configuration

Pada form HSS configuration di bagi 2 kelompok konfigurasi:

1. Global confguration

a) IP Address

i. *Value* dari IP Address diambil (load) dari parameter *host* yang terdapat di file konfigurasi hss.properties.

ii. Jika *value* IP Address diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok IP Address HSS akan secara otomatis diubah oleh sistem. Anggota kelompok IP Address HSS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Name

i. *Value* dari Name diambil (load) dari parameter *FQDN* yang terdapat di file konfigurasi DiameterPeerHss.xml.

ii. Jika *value* Name diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Nama HSS akan secara otomatis diubah oleh sistem. Anggota kelompok Nama HSS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

c) Diameter port

i. *Value* dari Diameter Port diambil (load) dari parameter *acceptor port* yang terdapat di file konfigurasi DiameterPeerHss.xml.

ii. Jika *value* Diameter Port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Diameter Port HSS akan secara otomatis diubah oleh sistem. Anggota kelompok Diameter Port HSS ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

d) Web server port

i. *Value* dari Web server port diambil (load) dari parameter *port* yang terdapat di file konfigurasi Hss.properties..

ii. Jika *value* Web server port diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok HSS port akan secara otomatis diubah oleh sistem. Anggota kelompok HSS port ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

2. Time

a) Subscription Expiry time

i. *Value* dari Subcription Expiry time diambil (load) dari parameter *expiry_time_lim* yang terdapat di file konfigurasi Hss.properties.

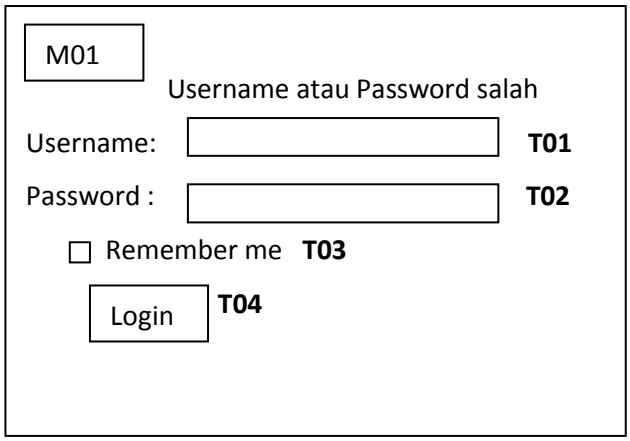
ii. Jika *value* Subcription Expiry diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Default Time Expiration of Subcription akan secara otomatis diubah oleh sistem. Anggota kelompok Default Time Expiration of Subcription ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

b) Dialogs expiration time

i. *Value* dari Dialogs expiration time diambil (load) dari parameter *dialogs_expiration_time* yang terdapat di file konfigurasi scscf.cfg

ii. Jika *value* Dialogs expiration time diubah dan kemudian disimpan, maka *value* dari seluruh anggota kelompok Default Time Expiration of Dialog akan secara otomatis diubah oleh sistem. Anggota kelompok Default Time Expiration of Dialog ini telah di definisikan di bahasan tentang analisis data pada sub bab 3.1.1.2 bagian pengelompokkan.

11. Pesan



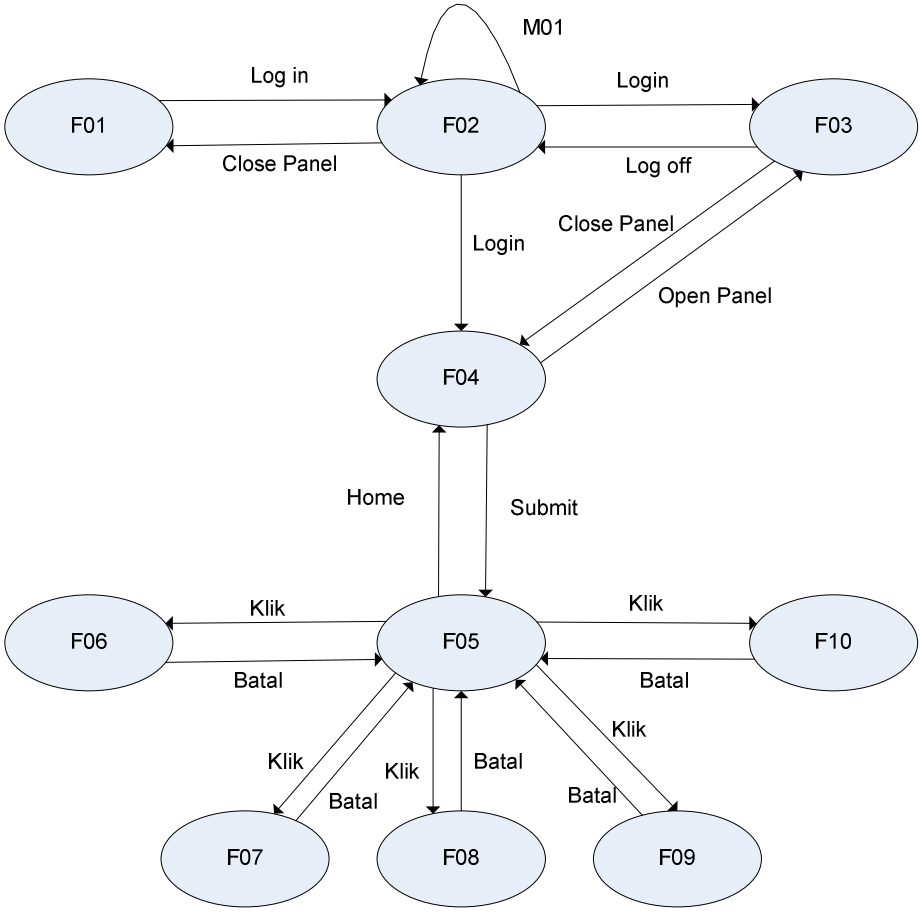
Gambar 3.16 Rancangan Tampilan Pesan Salah Password

3.2.2 Jaringan Semantik

Semua jaringan semantik dibangun dari dua komponen dasar, yaitu :

1. Node-node (simpul) yang merepresentasikan objek, konsep atau situasi. Node ditunjukkan atau diindikasikan oleh kotak atau lingkaran.
2. Arc (busur) merepresentasikan hubungan antara node-node. Arc ditunjukkan oleh panah yang menghubungkan node, dan menunjukkan link atau edge.

Berikut adalah jaringan semantik dari aplikasi yang akan dibangun:



Gambar 3.17 Jaringan Semantik

3.2.3 Modul dan algoritma

Pada aplikasi ini terdapat beberapa modul, yakni modul koneksi, modul login, modul edit user, modul delete user, modul simpan dan modul load. Modul koneksi berisikan algoritma untuk mengkoneksikan antara program dengan database. Modul login berisikan algoritma untuk login kedalam program konfigurasi *testbed* IMS, yakni dengan mencocokkan data user yang diinputkan dengan data user yang telah terdaftar pada database. Modul edit user berisikan algoritma untuk mengedit data user yang telah terdaftar di database. Modul delete user berisikan algoritma untuk menghapus user yang telah terdaftar di database. Modul load dan modul simpan merupakan modul utama yang menjadi inti aplikasi ini. Modul load berisikan algoritma pada saat aplikasi melakkukan baca suatu file konfigurasi dan mengambil value dari parameter yang dicari yang kemudian akan ditampilkan di form-form yang ada di aplikasi. Modul simpan berisikan algoritma pada saat melakkukan penyimpanan value yang telah diubah oleh admin. Algoritma dari modul load dan modul simpan ini selanjutnya dipaparkan pada bahasan berikut :

1. **Modul Load**

Algoritma Load Data Konfigurasi
{*Membaca dan mengambil value acuan dari file-file konfigurasi bedasarkan parameter dan valuenya*}

Deklarasi
Dir : string {nama file berikut directory dimana file konfigurasi berada}
Par : string {parameter untuk mengambil value}
Val : string {value yang diambil sesuai parameter}
Jum : integer {jumlah baris}

Deskripsi
Read(Dir) {baca nama file berikut directory dimana file konfigurasi berada}

```
If (file_exist(Dir)) then          {cek keberadaan file}
    Jum ← sizeof(Dir)        {cek jumlah baris}
    For i _ 0 to Jum          {baca baris sebanyak jum baris}
        If Explode('=',Dir)  then      { jika pisah dengan karakter
                                       pemisah samadengan}
            If explode[0] ← Par     {pengecekkan parameter}
                Val ← explode[1]   {tampung value}
                Write(Val)          {tampilkan value}
            Endif
        Else if  Explode(',',Dir)  then { pemisahan dengan tanda koma
             untuk              model modparam}
            If explode[1] ← Par        {pengecekkan parameter yang
                               berada di antara koma}
                Val ← explode[2]   {tampung value}
                Write(Val)          {tampilkan value}
            Endif
    Endfor
Else
   Write("File tidak ditemukan") {jika file tidak ditemukan}
Endif
```

2. Modul Simpan

```
Algoritma Simpan Data Konfigurasi
{menyimpan value, mengganti value lama di file-file konfigurasi
dengan value yang baru bedasarkan kelompok parameter dan
valuenya}

Deklarasi
Dir : string {nama file berikut directory dimana file konfigurasi
berada}
Par: string {parameter untuk mengambil value}
File: string {modul konfigurasi}
```

Val_lama : string {value lama yang diambil sesuai parameter}
Val_baru : string {value yang akan mengganti value lama sesuai parameter}

Deskripsi
Read(Dir) {baca file konfigurasi}
If file_exist(Dir) then {cek keberadaan file}
If (Val_lama != Val_baru) then {cek apakah value diganti atau tidak}
If (simpan global) then
{replace value lama dengan value baru}
Val_lama ← Val_baru
Elseif (simpan modparam) then
{replace value lama dengan value baru dengan pola modparam}
Replace (modparam("File","Par","val_lama") to modparam("File","Par","val_baru")
Elseif (simpan) then
{replace value lama dengan value baru dengan pola samadengan}
Replace ("Par"="val_lama") to ("Par"="val_baru")
Elseif (simpan comment) then
{replace " " dengan "#" pada baris pola samadengan}
Replace ("Par"="val_lama") to (#"Par"="val_baru")
Elseif (simpan comment modparam) then
{replace " " dengan "#" pada baris pola modparam}
Replace (modparam("File","Par","val_lama") to #modparam("File","Par","val_baru")
Elseif (simpan uncomment) then
{replace "#" dengan " " pada baris pola samadengan }
Replace (#"Par"="val_lama") to ("Par"="val_baru")
Elseif (simpan uncomment modparam) then
{replace " " dengan "#" pada baris pola modparam }
Replace (#modparam("File","Par","val_lama") to

```
            modparam("File","Par","val_baru")
        Endif
    Endif
Else
    Write("File tidak ditemukan") {jika file tidak ditemukan}
Endif
```

3.3 Implementasi

Tahap implementasi merupakan tahap penerjemahan dari tahap analisis kedalam bentuk yang sesungguhnya yang dapat dimengerti oleh oleh user.

3.3.1 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan adalah :

a) Microsoft windows 7.
b) Mysql sebagai database.
c) Adobe dreamweaver cs 4 untuk menulis kode program.
d) Adobe photoshop untuk mendesain tampilan.
e) Wamp server
f) Php.

3.3.2 Implementasi perangkat keras

Perangkat keras yang digunakan untuk pengoperasian Aplikasi pada TELKOM RDC dengan spesifikasi sebagai berikut :

a) Processor Intel Pentium IV 2,6 GHz
b) Monitor 14”
c) Hardisk Drive 80 Gb
d) Memory/RAM 1 Gb
e) VGA Card 256 Mb
f) CD ROM drive
g) Keyboard
h) Mouse

3.3.3 Implementasi database

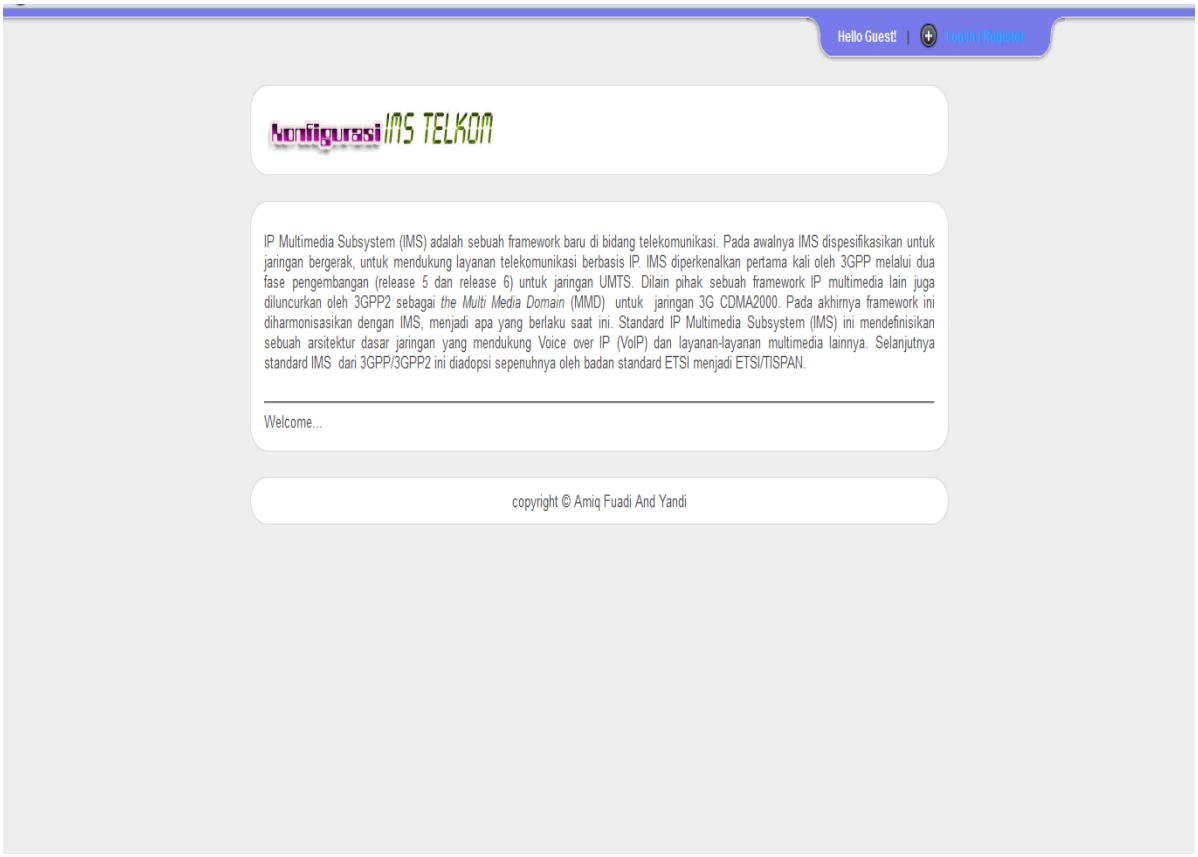
Database pada program ini digunakan untuk menyimpan data akses *right* siapa saja yang di ijinkan mengakses program oleh superadmin.File konfigurasi tidak dimasukan ke database, tapi masih dimasukan ke dalam badan program. Adapun isi dari databasenya adalah:

Nama field	Tipe	Ukuran
Id	Integer	2
Nama	Varchar	21
Nik	Varchar	11
Username	Varchar	11
Password	Varchar	32
Level	Varchar	11

Tabel 3. Database

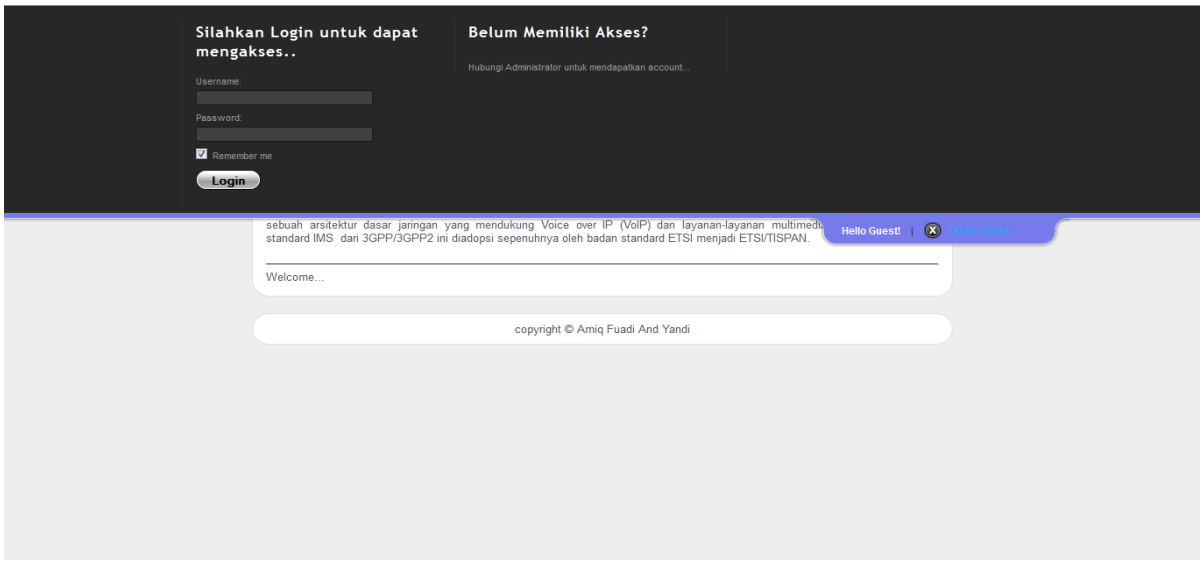
3.3.4 Implementasi Antar Muka

a) Tampilan Utama



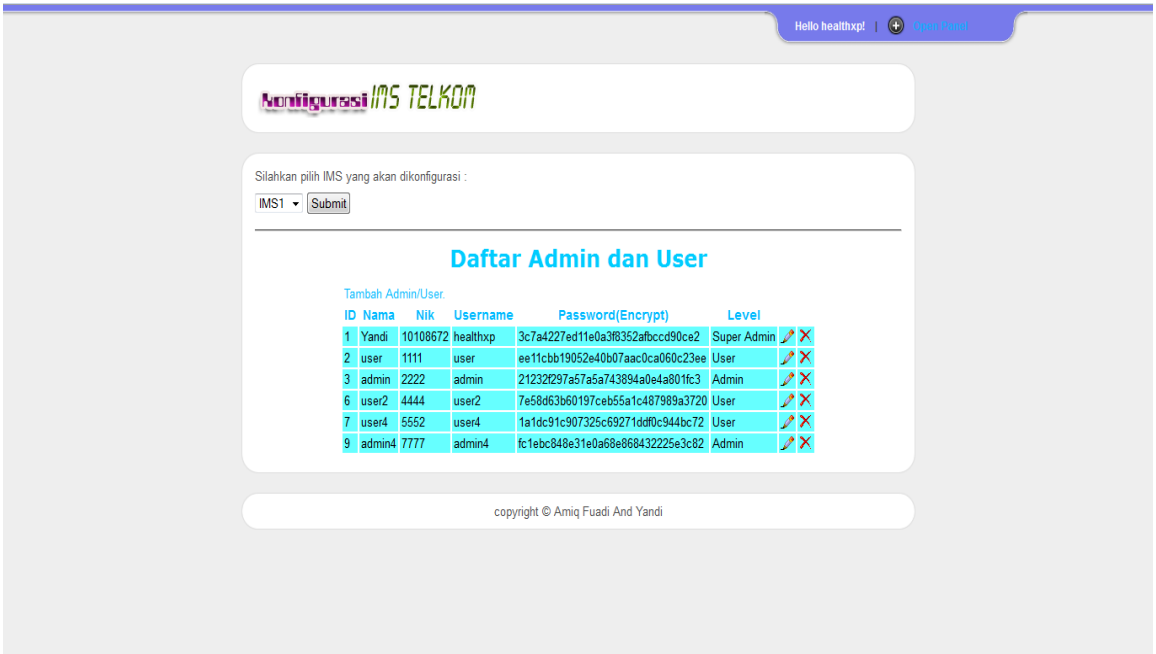
Gambar 3.18 tampilan utama

b) Tampilan Login



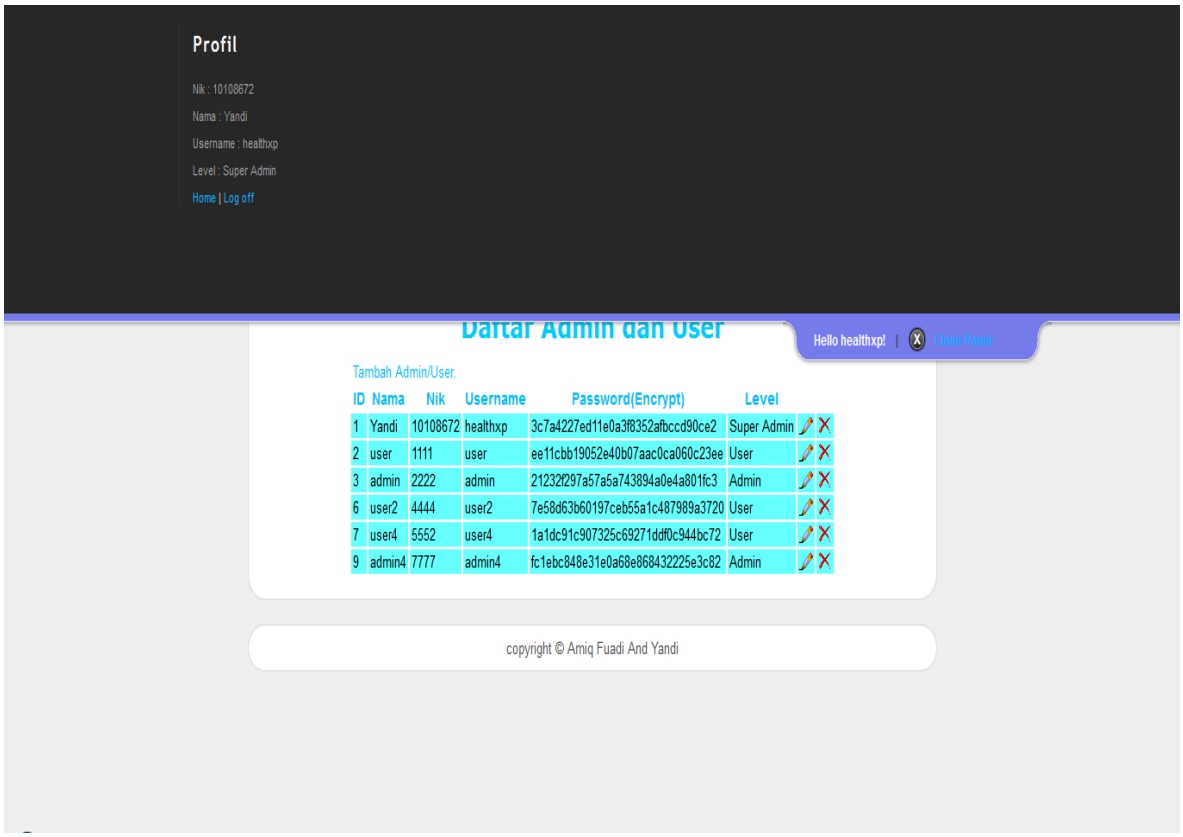
Gambar 3.19 tampilan login

c) Tampilan tabel admin dan user



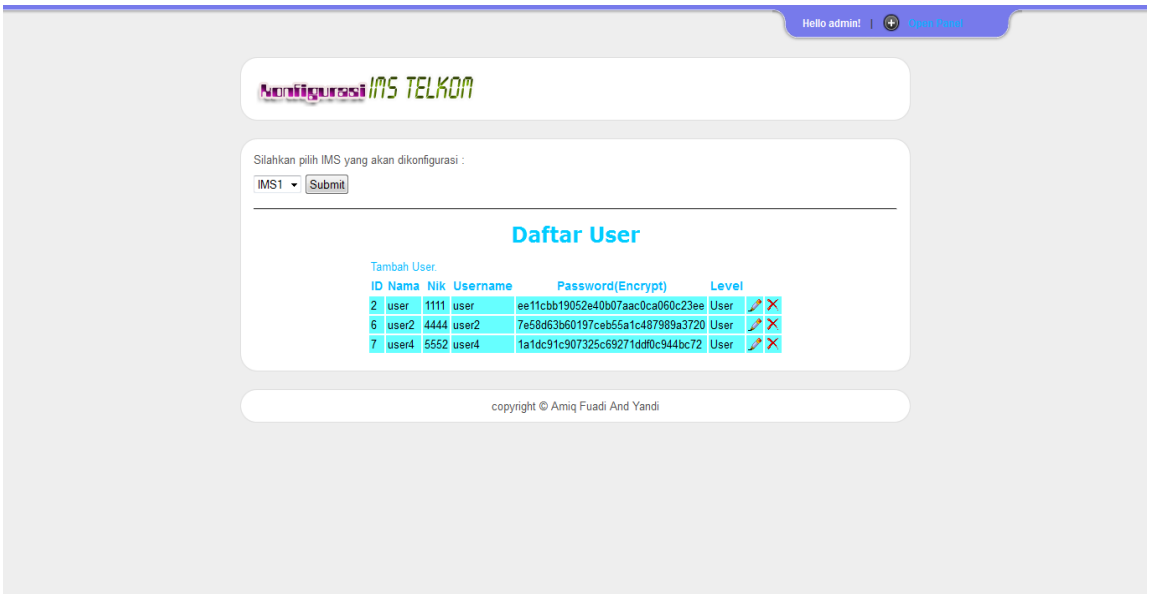
Gambar 3.20 tampilan tabel admin dan user

d) Tampilan profil user



Gambar 3.21 tampilan profil user

e) Tampilan daftar user



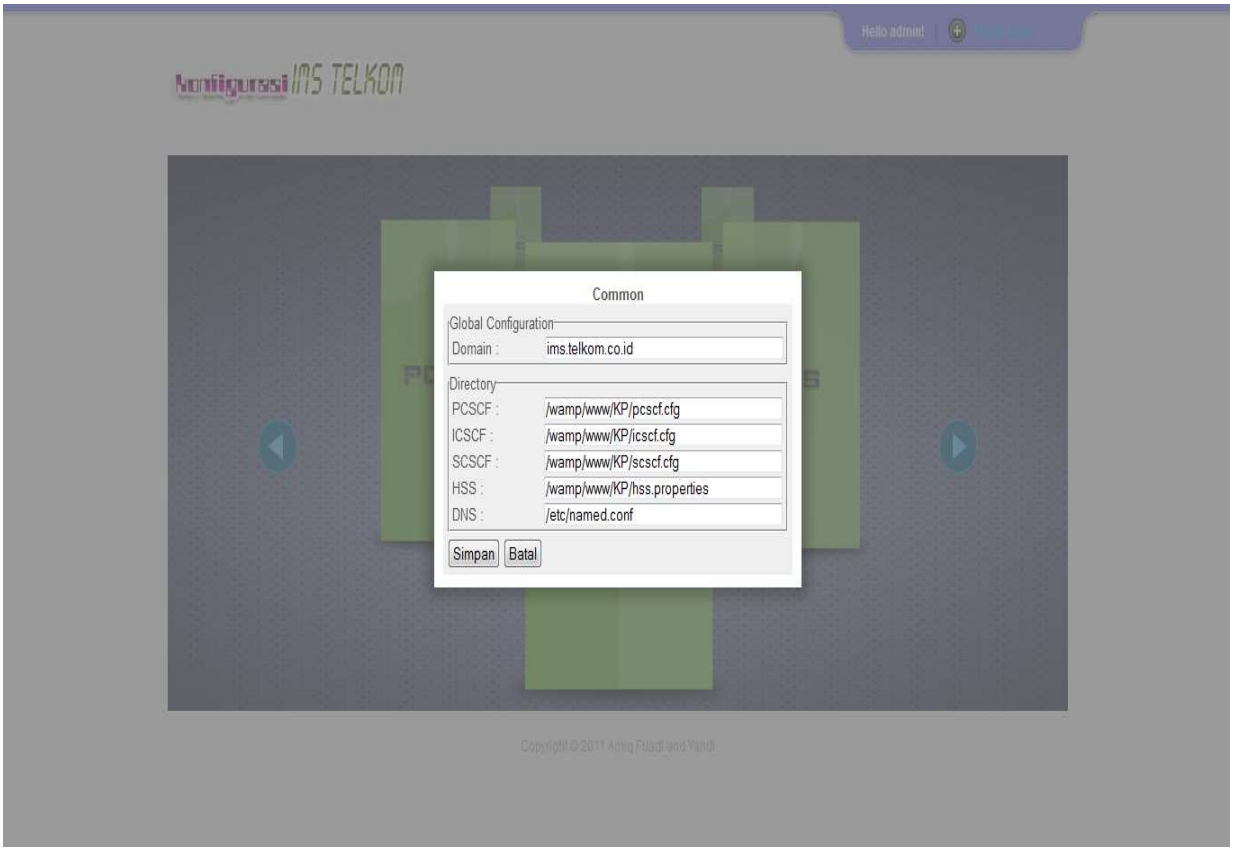
Gambar 3.22 tampilan daftar user

f) Tampilan konfigurasi



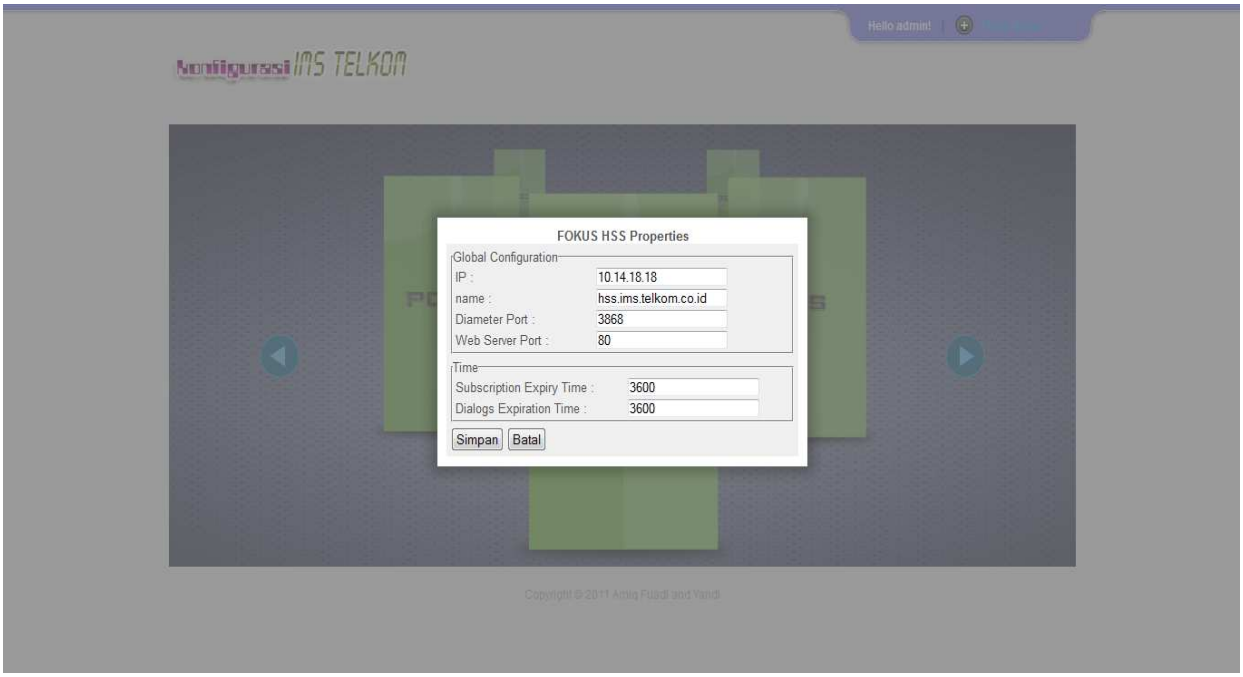
Gambar 3.23 tampilan konfigurasi

g) Tampilan common



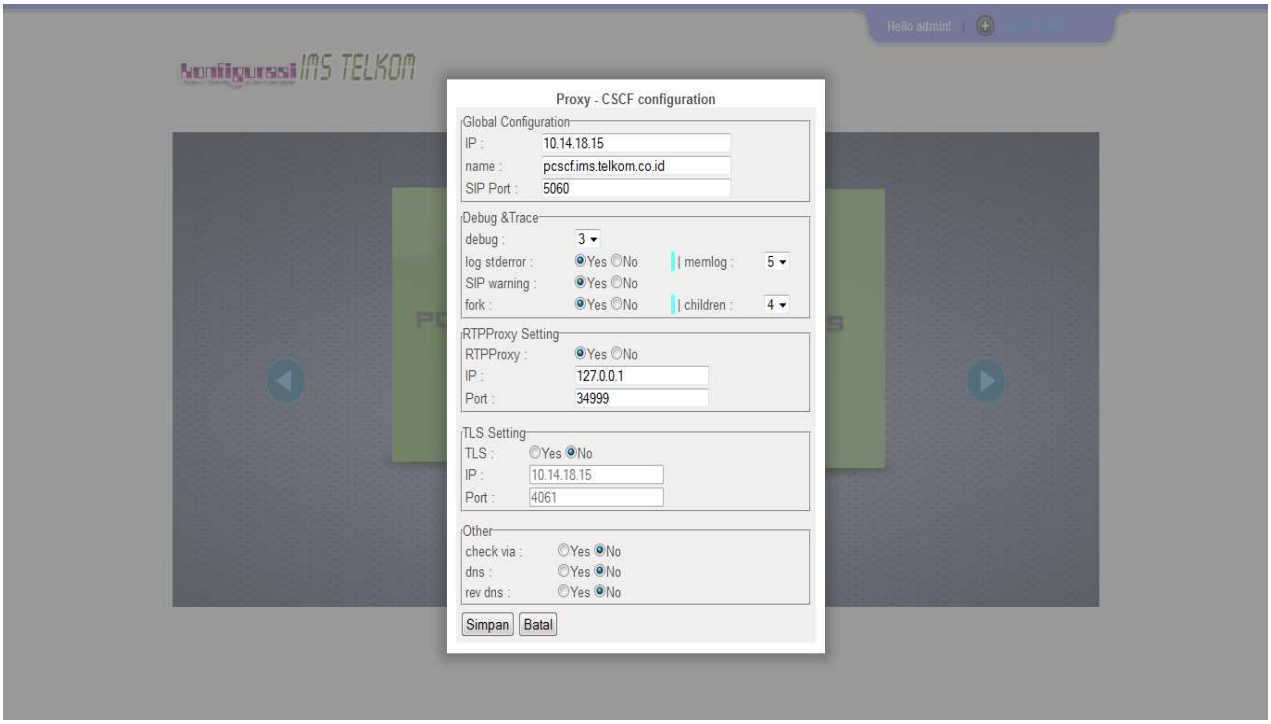
Gambar 3.24 tampilan common configuration

h) Tampilan HSS



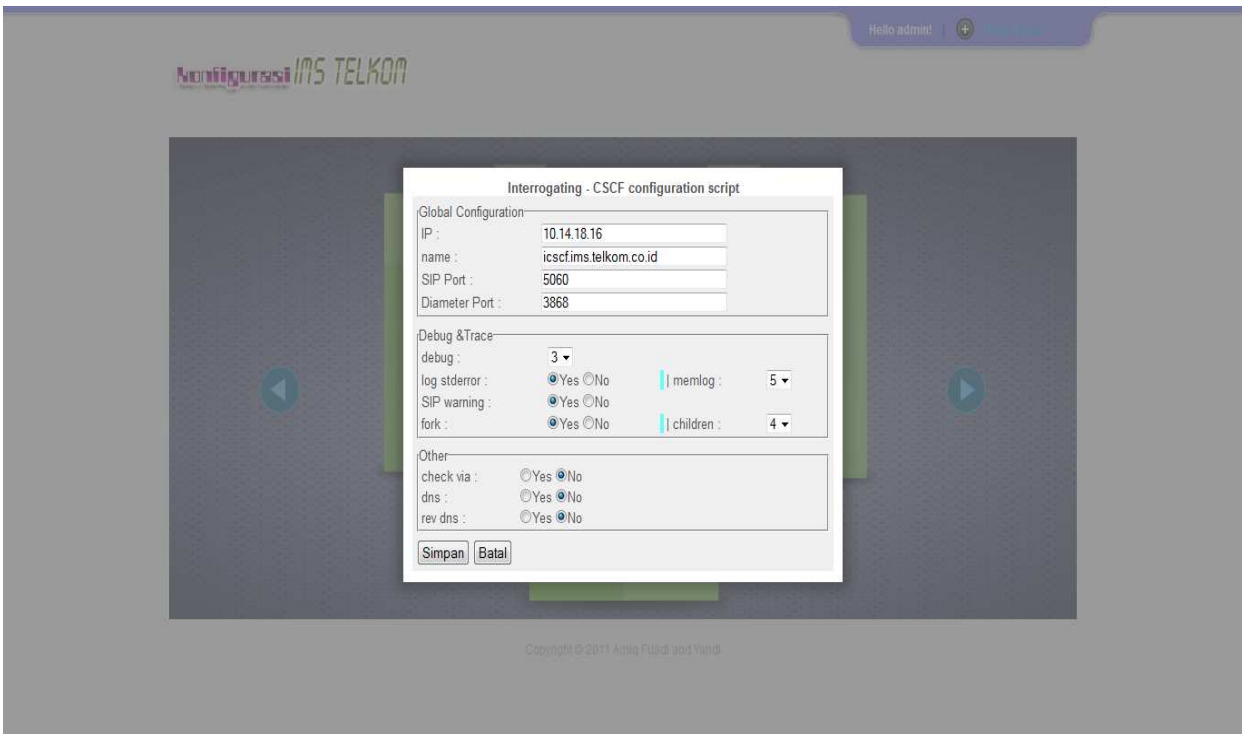
Gambar 3.25 Tampilan HSS configuration

i) Tampilan P-CSCF



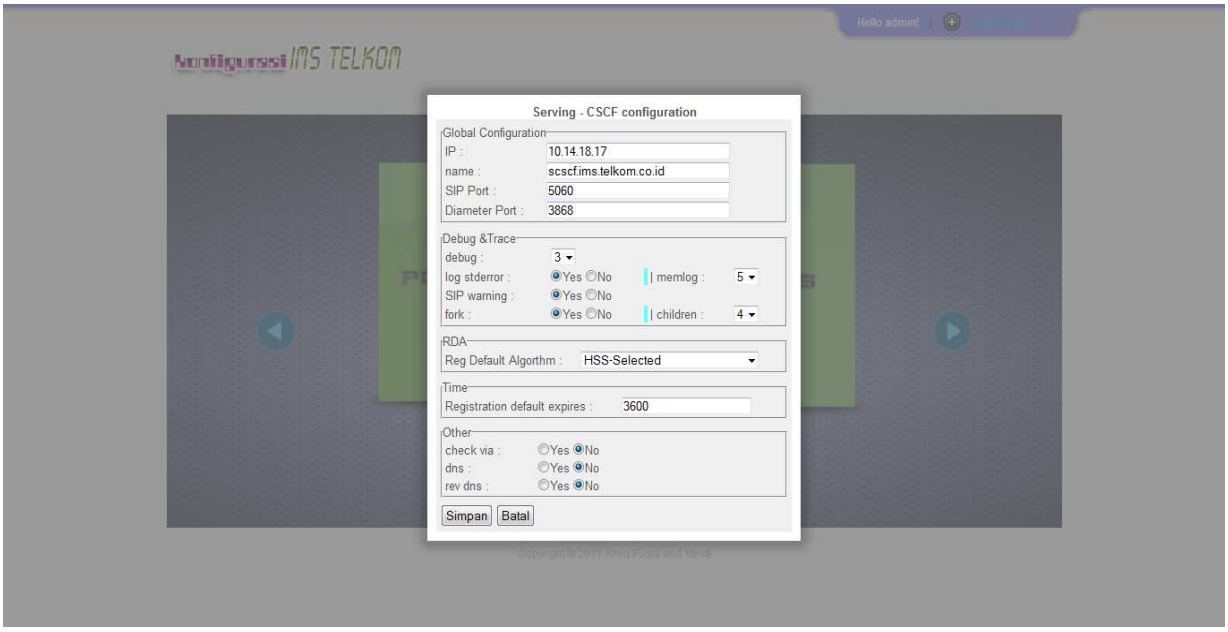
Gambar 3.26 tampilan P-CSCF configuration

j) Tampilan I-CSCF



Gambar 3.27 tampilan I-CSCF configuration

k) Tampilan S-CSCF



Gambar 3.28 tampilan S-CSCF configuration

3.4 Evaluasi Sistem

Pada masing-masing kolompok yang teridentifikasi, terdapat parameter dan value yang digunakan secara khusus (tanpa relasi dengan parameter dan value di file lain), maupun yang digunakan secara identik (memiliki relasi dengan parameter dan value di file lain). Value dapat dikenali dengan melihat parameternya, tetapi terdapat *anomaly* yang terjadi pada parameter alias yang terdapat di file icscf.cfg. Pada kasus ini terdapat dua parameter yang sama dengan value yang berbeda, sehingga sulit untuk diidentifikasi kepemilikan value yang berbeda kelompok ini. Maka dibuat suatu asumsi bahwa alias pertama pada baris merupakan alias dari nama I-CSCF dan alias berikutnya merupakan nama domain dari *testbed* IMS itu sendiri.

Pada sistem konfigurasi sebelumnya admin melakukkan konfigurasi secara manual dengan membuka dan mengedit file konfigurasi satu persatu dengan menggunakan text editor. Admin diharuskan teliti dalam mengedit dan mengetahui letak dan relasi value antar file yang akan diubah, dengan begitu potensi terjadi kesalahan cukup besar entah salah ketik, lupa atau terlewat tidak mengganti value yang padahal berelasi. Belum lagi waktu yang dibutuhkan dalam

konfigurasi akan cukup lama karena harus dilakukkan secara teliti dan manual. Dengan adanya aplikasi ini pada sistem, admin tidak perlu mengingat file-file konfigurasi berikut relasinya, diharapkan akan meminimalisir kesalahan dan mengefisienkan waktu saat konfigurasi, serta admin mendapatkan kemudahan dengan interface yang lebih baik.

Sebagai *prototype* aplikasi ini masih memiliki kekurangan. Pertama parameter-parameter yang dipakai dalam mengambil *value* disimpan di badan program dan tidak disimpan di database. Akibatnya, jika terdapat parameter baru yang ingin ditambahkan, maka harus dilakukan modifikasi program dengan cara membuka kode program dan menambahkan parameter tersebut secara manual pada badan program. Kedua, file konfigurasi harus berada di perangkat yang sama dengan aplikasi, akibatnya aplikasi harus ditanam di perangkat atau komputer dimana file konfigurasi berada.

Selain itu program masih menggunakan konsep pemrograman prosedural, sehingga apabila ingin menambahkan fitur atau fungsi baru pada program harus merubah program secara keseluruhan.