

LAMPIRAN 1

%Perhitungan Bandwith
 %Payload Paket Video dan Audio
 %Bcodecv1=128000;
 %Bcodecv2=256000;
 %Bcodecv3=384000;

Bcodecv4=128000;
 Bcodecv5=384000;
 Bcodecv0=[Bcodecv4:128000:Bcodecv5];

Bcodeca=48000;
 pb=0.00000001;
 fr=0.033;
 H_RTP=96;
 H_UDP=64;
 H_IP=320;
 max_PLv=2032;
 max_PLa=368;
 PLv1=Bcodecv0*fr;
 PLv2=Bcodecv0*fr;
 PLv3=Bcodecv0*fr;
 PLa=Bcodeca*fr;

%Jumlah Paket Video dan Audio

Pv1=PLv1/max_PLv;
 Pv2=PLv2/max_PLv;
 Pv3=PLv3/max_PLv;
 fix_Pv1=ceil(Pv1);
 fix_Pv2=ceil(Pv2);
 fix_Pv3=ceil(Pv3);
 Pa=PLa/max_PLa;
 fix_Pa=ceil(Pa);

%Besar Paket Video dan Audio

Pv_size1=PLv1 + (fix_Pv1*(H_RTP + H_UDP + H_IP));
 Pv_size2=PLv2 + (fix_Pv2*(H_RTP + H_UDP + H_IP));
 Pv_size3=PLv3 + (fix_Pv3*(H_RTP + H_UDP + H_IP));
 Pa_size=PLa + (fix_Pa*(H_RTP + H_UDP + H_IP));

%Besar Paket Video Streaming

Pvs_size1=Pv_size1 + Pa_size;
 Pvs_size2=Pv_size2 + Pa_size;
 Pvs_size3=Pv_size3 + Pa_size;
 Pvs_size_byte1=Pvs_size1/8;
 Pvs_size_byte2=Pvs_size2/8;
 Pvs_size_byte3=Pvs_size3/8;

%Bandwidth Video Streaming

Ba=Pa_size.*Bcodeca./PLa;
 Bv1=Pv_size1.*Bcodecv0./PLv1;
 Bv2=Pv_size2.*Bcodecv0./PLv2;
 Bv3=Pv_size3.*Bcodecv0./PLv3;

```

Bvs1=Ba+Bv1;
Bvs2=Ba+Bv2;
Bvs3=Ba+Bv3;
Bvs_kbps1=Bvs1/1000;
Bvs_kbps2=Bvs2/1000;
Bvs_kbps3=Bvs3/1000;

```

%Grafik

figure (1)

```

plot(Bcodecv0,Bvs_kbps1,'-^r', Bcodecv0,Bvs_kbps2,'-^r', Bcodecv0,Bvs_kbps3,'-^r')

```

grid on

```

title('Grafik Bandwidth Video Streaming Terhadap Bit Rate yang Berubah-ubah');

```

```

xlabel('Bit Rate Codec (bps)')

```

```

ylabel('Bandwidth Video Streaming (kbps)')

```

LAMPIRAN 2

% PERHITUNGAN DELAY PROSES ENKAPSULASI DAN DEKAPSULASI

```

%server

```

```

Pvs_size_byte1=1206;

```

```

Pvs_size_byte2=1854;

```

```

Pvs_size_byte3=2502;

```

```

MTU=1500;

```

```

header_eth=14;

```

```

header_UDP=8;

```

```

header_IP=40;

```

```

FCS=4;

```

```

C_server=10^9;

```

```

N_frame_eth_server1=Pvs_size_byte1/MTU;

```

```

N_frame_eth_server2=Pvs_size_byte2/MTU;

```

```

N_frame_eth_server3=Pvs_size_byte3/MTU;

```

```

W_frame_server1=Pvs_size_byte1+(2*(header_UDP+header_IP+header_eth+FCS));

```

```

W_frame_server2=Pvs_size_byte2+(2*(header_UDP+header_IP+header_eth+FCS));

```

```

W_frame_server3=Pvs_size_byte3+(2*(header_UDP+header_IP+header_eth+FCS));

```

```

te11=(W_frame_server1-Pvs_size_byte1)*8/C_server;

```

```

te12=(W_frame_server2-Pvs_size_byte2)*8/C_server;

```

```

te13=(W_frame_server3-Pvs_size_byte3)*8/C_server;

```

```

%PDN-GW

```

```

header_GTP=8;

```

```

C_PDNGW=10^8;

```

```

FCS=4;

```

```

W_data_PDNGW1=W_frame_server1-(2*(header_UDP+header_IP+header_eth+FCS));

```

```

W_data_PDNGW2=W_frame_server2-(2*(header_UDP+header_IP+header_eth+FCS));

```

```

W_data_PDNGW3=W_frame_server3-(2*(header_UDP+header_IP+header_eth+FCS));

```

```

td1=(W_frame_server1-W_data_PDNGW1)*8/C_PDNGW;

```

```

td2=(W_frame_server2-W_data_PDNGW2)*8/C_PDNGW;

```

```

td3=(W_frame_server3-W_data_PDNGW3)*8/C_PDNGW;

```

```

MSS=MTU-header_GTP-header_UDP-header_IP;

```

```

N_dtgrm_PDNGW1=W_data_PDNGW1/MSS;

```

```

N_dtgrm_PDNGW2=W_data_PDNGW2/MSS;

```

```

N_dtgrm_PDNGW3=W_data_PDNGW3/MSS;

```

```

W_dtgrm_PDNGW1=W_data_PDNGW1+(2*(header_GTP+header_UDP+header_IP));

```

```

W_dtgrm_PDNGW2=W_data_PDNGW2+(2*(header_GTP+header_UDP+header_IP));
W_dtgrm_PDNGW3=W_data_PDNGW3+(2*(header_GTP+header_UDP+header_IP));
N_frame_eth_PDNGW1=W_dtgrm_PDNGW1/MTU;
N_frame_eth_PDNGW2=W_dtgrm_PDNGW2/MTU;
N_frame_eth_PDNGW3=W_dtgrm_PDNGW3/MTU;
W_frame_PDNGW1=W_dtgrm_PDNGW1+(2*(header_eth+FCS));
W_frame_PDNGW2=W_dtgrm_PDNGW2+(2*(header_eth+FCS));
W_frame_PDNGW3=W_dtgrm_PDNGW3+(2*(header_eth+FCS));
te21=(W_frame_PDNGW1-W_data_PDNGW1)*8/C_PDNGW;
te22=(W_frame_PDNGW2-W_data_PDNGW2)*8/C_PDNGW;
te23=(W_frame_PDNGW3-W_data_PDNGW3)*8/C_PDNGW;

```

```
%SGW
```

```
C_SGW=10^8;
```

```
FCS=4;
```

```
W_data_SGW1=W_frame_PDNGW1-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW2=W_frame_PDNGW2-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW3=W_frame_PDNGW3-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
td21=(W_frame_PDNGW1-W_data_SGW1)*8/C_SGW;
```

```
td22=(W_frame_PDNGW2-W_data_SGW2)*8/C_SGW;
```

```
td23=(W_frame_PDNGW3-W_data_SGW3)*8/C_SGW;
```

```
N_dtgrm_SGW1=W_data_SGW1/MSS;
```

```
N_dtgrm_SGW2=W_data_SGW2/MSS;
```

```
N_dtgrm_SGW3=W_data_SGW3/MSS;
```

```
W_dtgrm_SGW1=W_data_SGW1+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW2=W_data_SGW2+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW3=W_data_SGW3+(2*(header_GTP+header_UDP+header_IP));
```

```
N_frame_eth_SGW1=W_dtgrm_SGW1/MTU;
```

```
N_frame_eth_SGW2=W_dtgrm_SGW2/MTU;
```

```
N_frame_eth_SGW3=W_dtgrm_SGW3/MTU;
```

```
W_frame_SGW1=W_dtgrm_SGW1+(2*(header_eth+FCS));
```

```
W_frame_SGW2=W_dtgrm_SGW2+(2*(header_eth+FCS));
```

```
W_frame_SGW3=W_dtgrm_SGW3+(2*(header_eth+FCS));
```

```
te31=(W_frame_SGW1-W_data_SGW1)*8/C_SGW;
```

```
te32=(W_frame_SGW2-W_data_SGW2)*8/C_SGW;
```

```
te33=(W_frame_SGW3-W_data_SGW3)*8/C_SGW;
```

```
%eNB
```

```
C_eNB=155.52*(10^6);
```

```
header_PDCP=1;
```

```
header_MAC=0.5;
```

```
header_RLC=2;
```

```
W_data_eNB1=W_frame_SGW1-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_eNB2=W_frame_SGW2-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_eNB3=W_frame_SGW3-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
td31=(W_frame_SGW1-W_data_eNB1)*8/C_eNB;
```

```
td32=(W_frame_SGW2-W_data_eNB2)*8/C_eNB;
```

```
td33=(W_frame_SGW3-W_data_eNB3)*8/C_eNB;
```

```
N_dtgrm_eNB1=W_data_eNB1/MTU;
```

```

N_dtgrm_eNB2=W_data_eNB2/MTU;
N_dtgrm_eNB3=W_data_eNB3/MTU;
W_frame_PDCP1=W_data_eNB1+header_PDCP;
W_frame_PDCP2=W_data_eNB2+header_PDCP;
W_frame_PDCP3=W_data_eNB3+header_PDCP;
N_frame_RLC1=W_frame_PDCP1/40;
N_frame_RLC2=W_frame_PDCP2/40;
N_frame_RLC3=W_frame_PDCP3/40;
W_frame_RLCtot1=31*42;
W_frame_RLCtot2=47*42;
W_frame_RLCtot3=63*42;
N_frame_MAC1=W_frame_RLCtot1/(40+header_RLC);
N_frame_MAC2=W_frame_RLCtot2/(40+header_RLC);
N_frame_MAC3=W_frame_RLCtot3/(40+header_RLC);
W_frame_MAC=42+header_MAC;
W_frame_eNB1=W_frame_MAC*31;
W_frame_eNB2=W_frame_MAC*47;
W_frame_eNB3=W_frame_MAC*63;
te41=(W_frame_eNB1-W_data_eNB1)*8/C_eNB;
te42=(W_frame_eNB2-W_data_eNB2)*8/C_eNB;
te43=(W_frame_eNB3-W_data_eNB3)*8/C_eNB;

```

```
%UE
```

```

C_UE=3.6*(10^8);
W_data_UE1=W_frame_eNB1-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
W_data_UE2=W_frame_eNB2-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
W_data_UE3=W_frame_eNB3-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
td41=(W_frame_eNB1-W_data_UE1)*8/C_UE;
td42=(W_frame_eNB2-W_data_UE2)*8/C_UE;
td43=(W_frame_eNB3-W_data_UE3)*8/C_UE;

```

```

te_tot1=te11+te21+te31+te41;
te_tot2=te12+te22+te32+te42;
te_tot3=te13+te23+te33+te43;
td_tot1=td1+td21+td31+td41;
td_tot2=td2+td22+td32+td42;
td_tot3=td3+td23+td33+td43;

```

```
%PERHITUNGAN DELAY TRANSMISI
```

```

cr=3.84*(10^6);
rf1=0.01;
tt11=W_frame_server1*8/C_server;
tt12=W_frame_server2*8/C_server;
tt13=W_frame_server3*8/C_server;
tt21=W_frame_PDNGW1*8/C_PDNGW;
tt22=W_frame_PDNGW2*8/C_PDNGW;
tt23=W_frame_PDNGW3*8/C_PDNGW;
tt31=W_frame_SGW1*8/C_SGW;
tt32=W_frame_SGW2*8/C_SGW;

```

```

tt33=W_frame_SGW3*8/C_SGW;
tt41=(W_frame_eNB1*8)/((cr*rf1/2560)*C_eNB);
tt42=(W_frame_eNB2*8)/((cr*rf1/2560)*C_eNB);
tt43=(W_frame_eNB3*8)/((cr*rf1/2560)*C_eNB);
tt_tot1=tt11+tt21+tt31+tt41;
tt_tot2=tt12+tt22+tt32+tt42;
tt_tot3=tt13+tt23+tt33+tt43;

```

%PERHITUNGAN DELAY PROPAGASI

```

d_server_PDNGW=5000;
d_PDNGW_SGW=5000;
d_SGW_eNB=5000;
d_eNB_UE1=1000;

```

```

c=3*10^8;
tp1=2*d_server_PDNGW/c;
tp2=2*d_PDNGW_SGW/c;
tp3=2*d_SGW_eNB/c;
tp41=31*d_eNB_UE1/c;
tp42=47*d_eNB_UE1/c;
tp43=63*d_eNB_UE1/c;
tp_tot1=tp1+tp2+tp3+tp41;
tp_tot2=tp1+tp2+tp3+tp42;
tp_tot3=tp1+tp2+tp3+tp43;

```

%PERHITUNGAN DELAY ANTRIAN

```

miu_PDNGW1=C_PDNGW/(W_frame_PDNGW1*8);
miu_PDNGW2=C_PDNGW/(W_frame_PDNGW2*8);
miu_PDNGW3=C_PDNGW/(W_frame_PDNGW3*8);
miu_SGW1=C_SGW/(W_frame_SGW1*8);
miu_SGW2=C_SGW/(W_frame_SGW2*8);
miu_SGW3=C_SGW/(W_frame_SGW3*8);
miu_eNB1=C_eNB/(W_frame_eNB1*8);
miu_eNB2=C_eNB/(W_frame_eNB2*8);
miu_eNB3=C_eNB/(W_frame_eNB3*8);
ro0=0.1;
ro1=0.9;
ro=[ro0:0.1:ro1];
lamda_PDNGW1=miu_PDNGW1.*ro;
lamda_PDNGW2=miu_PDNGW2.*ro;
lamda_PDNGW3=miu_PDNGW3.*ro;
lamda_SGW1=miu_SGW1.*ro;
lamda_SGW2=miu_SGW2.*ro;
lamda_SGW3=miu_SGW3.*ro;
lamda_eNB1=miu_eNB1.*ro;
lamda_eNB2=miu_eNB2.*ro;
lamda_eNB3=miu_eNB3.*ro;
tw11=lamda_PDNGW1./(miu_PDNGW1*(miu_PDNGW1-
lamda_PDNGW1))+1/miu_PDNGW1;
tw12=lamda_PDNGW2./(miu_PDNGW2*(miu_PDNGW2-
lamda_PDNGW2))+1/miu_PDNGW2;

```

```

tw13=lamda_PDNGW3./(miu_PDNGW3*(miu_PDNGW3-
lamda_PDNGW3))+(1/miu_PDNGW3);
tw21=lamda_SGW1./(miu_SGW1*(miu_SGW1-lamda_SGW1))+(1/miu_SGW1);
tw22=lamda_SGW2./(miu_SGW2*(miu_SGW2-lamda_SGW2))+(1/miu_SGW2);
tw23=lamda_SGW3./(miu_SGW3*(miu_SGW3-lamda_SGW3))+(1/miu_SGW3);
tw31=lamda_eNB1./(miu_eNB1*(miu_eNB1-lamda_eNB1))+(1/miu_eNB1);
tw32=lamda_eNB2./(miu_eNB2*(miu_eNB2-lamda_eNB2))+(1/miu_eNB2);
tw33=lamda_eNB3./(miu_eNB3*(miu_eNB3-lamda_eNB3))+(1/miu_eNB3);
tw_tot1=tw11+tw21+tw31;
tw_tot2=tw12+tw22+tw32;
tw_tot3=tw13+tw23+tw33;

```

```
%PERHITUNGAN DELAY CODEC
```

```

ta=0.04;
tv=0.3;
t_CODEC=ta+tv;

```

```
%PERHITUNGAN DELAY TOTAL
```

```

t_tot1=t_CODEC+te_tot1+td_tot1+tt_tot1+tp_tot1+tw_tot1;
t_tot2=t_CODEC+te_tot2+td_tot2+tt_tot2+tp_tot2+tw_tot1;
t_tot3=t_CODEC+te_tot3+td_tot3+tt_tot3+tp_tot3+tw_tot3;

```

```
%GRAFIK
```

```

figure (1)
plot(ro,t_tot1,'-r',ro,t_tot2,'-b',ro,t_tot3,'-g')
grid on
title('Grafik Delay End-to-End Terhadap Faktor Utilisasi Dengan Bit Rate yang Berubah-ubah');
legend('Bit Rate=128 kbps','Bit Rate=256 kbps','Bit Rate=384 kbps')
xlabel ('Faktor Utilisasi')
ylabel ('Delay End-to-End (s)')

```

LAMPIRAN 3

```
% PERHITUNGAN PROBABILITAS PACKET LOSS
```

```

%server
Pvs_size_byte1=1206;
Pvs_size_byte2=1854;
Pvs_size_byte3=2502;
MTU=1500;
header_eth=14;
header_UDP=8;
header_IP=40;
FCS=4;
C_server=10^9;
N_frame_eth_server1=Pvs_size_byte1/MTU;
N_frame_eth_server2=Pvs_size_byte2/MTU;
N_frame_eth_server3=Pvs_size_byte3/MTU;
W_frame_server1=Pvs_size_byte1+(2*(header_UDP+header_IP+header_eth+FCS));
W_frame_server2=Pvs_size_byte2+(2*(header_UDP+header_IP+header_eth+FCS));
W_frame_server3=Pvs_size_byte3+(2*(header_UDP+header_IP+header_eth+FCS));

```

```
te11=(W_frame_server1-Pvs_size_byte1)*8/C_server;
te12=(W_frame_server2-Pvs_size_byte2)*8/C_server;
te13=(W_frame_server3-Pvs_size_byte3)*8/C_server;
```

```
%PDN-GW
```

```
header_GTP=8;
```

```
C_PDNGW=10^8;
```

```
FCS=4;
```

```
W_data_PDNGW1=W_frame_server1-(2*(header_UDP+header_IP+header_eth+FCS));
```

```
W_data_PDNGW2=W_frame_server2-(2*(header_UDP+header_IP+header_eth+FCS));
```

```
W_data_PDNGW3=W_frame_server3-(2*(header_UDP+header_IP+header_eth+FCS));
```

```
td1=(W_frame_server1-W_data_PDNGW1)*8/C_PDNGW;
```

```
td2=(W_frame_server2-W_data_PDNGW2)*8/C_PDNGW;
```

```
td3=(W_frame_server3-W_data_PDNGW3)*8/C_PDNGW;
```

```
MSS=MTU-header_GTP-header_UDP-header_IP;
```

```
N_dtgrm_PDNGW1=W_data_PDNGW1/MSS;
```

```
N_dtgrm_PDNGW2=W_data_PDNGW2/MSS;
```

```
N_dtgrm_PDNGW3=W_data_PDNGW3/MSS;
```

```
W_dtgrm_PDNGW1=W_data_PDNGW1+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_PDNGW2=W_data_PDNGW2+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_PDNGW3=W_data_PDNGW3+(2*(header_GTP+header_UDP+header_IP));
```

```
N_frame_eth_PDNGW1=W_dtgrm_PDNGW1/MTU;
```

```
N_frame_eth_PDNGW2=W_dtgrm_PDNGW2/MTU;
```

```
N_frame_eth_PDNGW3=W_dtgrm_PDNGW3/MTU;
```

```
W_frame_PDNGW1=W_dtgrm_PDNGW1+(2*(header_eth+FCS));
```

```
W_frame_PDNGW2=W_dtgrm_PDNGW2+(2*(header_eth+FCS));
```

```
W_frame_PDNGW3=W_dtgrm_PDNGW3+(2*(header_eth+FCS));
```

```
te21=(W_frame_PDNGW1-W_data_PDNGW1)*8/C_PDNGW;
```

```
te22=(W_frame_PDNGW2-W_data_PDNGW2)*8/C_PDNGW;
```

```
te23=(W_frame_PDNGW3-W_data_PDNGW3)*8/C_PDNGW;
```

```
%SGW
```

```
C_SGW=10^8;
```

```
FCS=4;
```

```
W_data_SGW1=W_frame_PDNGW1-  
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW2=W_frame_PDNGW2-  
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW3=W_frame_PDNGW3-  
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
td21=(W_frame_PDNGW1-W_data_SGW1)*8/C_SGW;
```

```
td22=(W_frame_PDNGW2-W_data_SGW2)*8/C_SGW;
```

```
td23=(W_frame_PDNGW3-W_data_SGW3)*8/C_SGW;
```

```
N_dtgrm_SGW1=W_data_SGW1/MSS;
```

```
N_dtgrm_SGW2=W_data_SGW2/MSS;
```

```
N_dtgrm_SGW3=W_data_SGW3/MSS;
```

```
W_dtgrm_SGW1=W_data_SGW1+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW2=W_data_SGW2+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW3=W_data_SGW3+(2*(header_GTP+header_UDP+header_IP));
```

```
N_frame_eth_SGW1=W_dtgrm_SGW1/MTU;
```

```
N_frame_eth_SGW2=W_dtgrm_SGW2/MTU;
```

```
N_frame_eth_SGW3=W_dtgrm_SGW3/MTU;
```

```
W_frame_SGW1=W_dtgrm_SGW1+(2*(header_eth+FCS));
```

```
W_frame_SGW2=W_dtgrm_SGW2+(2*(header_eth+FCS));
```

```
W_frame_SGW3=W_dtgrm_SGW3+(2*(header_eth+FCS));
```

```
te31=(W_frame_SGW1-W_data_SGW1)*8/C_SGW;
te32=(W_frame_SGW2-W_data_SGW2)*8/C_SGW;
te33=(W_frame_SGW3-W_data_SGW3)*8/C_SGW;
```

```
%eNB
```

```
C_eNB=155.52*(10^6);
```

```
header_PDCP=1;
```

```
header_MAC=0.5;
```

```
header_RLC=2;
```

```
W_data_eNB1=W_frame_SGW1-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_eNB2=W_frame_SGW2-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_eNB3=W_frame_SGW3-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
td31=(W_frame_SGW1-W_data_eNB1)*8/C_eNB;
```

```
td32=(W_frame_SGW2-W_data_eNB2)*8/C_eNB;
```

```
td33=(W_frame_SGW3-W_data_eNB3)*8/C_eNB;
```

```
N_dtgrm_eNB1=W_data_eNB1/MTU;
```

```
N_dtgrm_eNB2=W_data_eNB2/MTU;
```

```
N_dtgrm_eNB3=W_data_eNB3/MTU;
```

```
W_frame_PDCP1=W_data_eNB1+header_PDCP;
```

```
W_frame_PDCP2=W_data_eNB2+header_PDCP;
```

```
W_frame_PDCP3=W_data_eNB3+header_PDCP;
```

```
N_frame_RLC1=W_frame_PDCP1/40;
```

```
N_frame_RLC2=W_frame_PDCP2/40;
```

```
N_frame_RLC3=W_frame_PDCP3/40;
```

```
W_frame_RLCtot1=31*42;
```

```
W_frame_RLCtot2=47*42;
```

```
W_frame_RLCtot3=63*42;
```

```
N_frame_MAC1=W_frame_RLCtot1/(40+header_RLC);
```

```
N_frame_MAC2=W_frame_RLCtot2/(40+header_RLC);
```

```
N_frame_MAC3=W_frame_RLCtot3/(40+header_RLC);
```

```
W_frame_MAC=42+header_MAC;
```

```
W_frame_eNB1=W_frame_MAC*31;
```

```
W_frame_eNB2=W_frame_MAC*47;
```

```
W_frame_eNB3=W_frame_MAC*63;
```

```
te41=(W_frame_eNB1-W_data_eNB1)*8/C_eNB;
```

```
te42=(W_frame_eNB2-W_data_eNB2)*8/C_eNB;
```

```
te43=(W_frame_eNB3-W_data_eNB3)*8/C_eNB;
```

```
%Payload Paket Video dan Audio
```

```
%Bcodecv1=128000;
```

```
%Bcodecv2=256000;
```

```
%Bcodecv3=384000;
```

```
Bcodecv4=128000;
```

```
Bcodecv5=384000;
```

```
Bcodecv0=[Bcodecv4:128000:Bcodecv5];
```

```
Bcodeca=48000;
```

```
pb=0.00000001;
```

```
fr=0.033;
```

```
H_RTP=96;
```

```
H_UDP=64;
```



```
H_IP=320;
max_PLv=2032;
max_PLA=368;
PLv1=Bcodecv0*fr;
PLv2=Bcodecv0*fr;
PLv3=Bcodecv0*fr;
PLA=Bcodeca*fr;
```

```
%Jumlah Paket Video dan Audio
```

```
Pv1=PLv1/max_PLv;
Pv2=PLv2/max_PLv;
Pv3=PLv3/max_PLv;
fix_Pv1=ceil(Pv1);
fix_Pv2=ceil(Pv2);
fix_Pv3=ceil(Pv3);
Pa=PLA/max_PLA;
fix_Pa=ceil(Pa);
```

```
%Besar Paket Video dan Audio
```

```
Pv_size1=PLv1 + (fix_Pv1*(H_RTP + H_UDP + H_IP));
Pv_size2=PLv2 + (fix_Pv2*(H_RTP + H_UDP + H_IP));
Pv_size3=PLv3 + (fix_Pv3*(H_RTP + H_UDP + H_IP));
Pa_size=PLA + (fix_Pa*(H_RTP + H_UDP + H_IP));
```

```
%Besar Paket Video Streaming
```

```
Pvs_size1=Pv_size1 + Pa_size;
Pvs_size2=Pv_size2 + Pa_size;
Pvs_size3=Pv_size3 + Pa_size;
Pvs_size_byte1=Pvs_size1/8;
Pvs_size_byte2=Pvs_size2/8;
Pvs_size_byte3=Pvs_size3/8;
```

```
%packet loss server
```

```
Pb=10^-8;
p_VS1=Pvs_size1*pb;
p_VS2=Pvs_size2*pb;
p_VS3=Pvs_size3*pb;
```

```
%BIT ERROR RATE
```

```
PG=12;
NF=7;
Pt=46;
Lt=2;
Lr=0;
Gt=18;
Gr=0;
K=1.381*10^-23;
T=300;
B=10*10^6;
R=3.6*10^6;
c=3*10^8;
f=2.310^9;
d=1000;
M=64;
No=10*log10(K*T)+10*log10(B)+NF;
lamda=c/f;
```

```

FSL=20*log10((4*pi*d)/lamda);
Pr=Pt-FSL-Lt-Lr-Gr-Gt;
SNR=Pr-No;
Eb_per_No=SNR-10*log10(B/R);
x=sqrt((3*Eb_per_No*log2(M))/(M-1));
BER=((6*(sqrt(M)-1))/(sqrt(M))*log2(M))*(0.5*erfc(x/sqrt(2)));

```

```

%packet loss LTE
p_LTE1=(W_frame_eNB1*8)*BER;
p_LTE2=(W_frame_eNB2*8)*BER;
p_LTE3=(W_frame_eNB3*8)*BER;

```

```

%total packet loss
p_tot1=1-((1-p_LTE1)*(1-p_VS1));
p_tot2=1-((1-p_LTE2)*(1-p_VS2));
p_tot3=1-((1-p_LTE3)*(1-p_VS3));

```

```

%grafik
plot(Bcodecv0,p_tot1,'-^r',Bcodecv0,p_tot2,'-^r',Bcodecv0,p_tot3,'-^r')
grid on
title('Grafik Probabilitas Packet Loss Terhadap Bit Rate yang Berubah-ubah');
xlabel('Bit Rate Codec')
ylabel('Probabilitas Packet Loss')

```

LAMPIRAN 4

```

%Payload Paket Video dan Audio

```

```

Bcodecv1=128000;
Bcodecv2=256000;
Bcodecv3=384000;
Bcodeca=48000;
pb=0.00000001;
fr=0.033;
H_RTP=96;
H_UDP=64;
H_IP=320;
max_PLv=2032;
max_PLa=368;
PLv1=Bcodecv1*fr;
PLv2=Bcodecv2*fr;
PLv3=Bcodecv3*fr;
PLa=Bcodeca*fr;

```

```

%Jumlah Paket Video dan Audio

```

```

Pv1=PLv1/max_PLv;
Pv2=PLv2/max_PLv;
Pv3=PLv3/max_PLv;
fix_Pv1=ceil(Pv1);
fix_Pv2=ceil(Pv2);
fix_Pv3=ceil(Pv3);
Pa=PLa/max_PLa;
fix_Pa=ceil(Pa);

```

% Besar Paket Video dan Audio

$Pv_size1 = PLv1 + (fix_Pv1 * (H_RTP + H_UDP + H_IP));$

$Pv_size2 = PLv2 + (fix_Pv2 * (H_RTP + H_UDP + H_IP));$

$Pv_size3 = PLv3 + (fix_Pv3 * (H_RTP + H_UDP + H_IP));$

$Pa_size = PLa + (fix_Pa * (H_RTP + H_UDP + H_IP));$

% Besar Paket Video Streaming

$Pvs_size1 = Pv_size1 + Pa_size;$

$Pvs_size2 = Pv_size2 + Pa_size;$

$Pvs_size3 = Pv_size3 + Pa_size;$

$Pvs_size_byte1 = Pvs_size1 / 8;$

$Pvs_size_byte2 = Pvs_size2 / 8;$

$Pvs_size_byte3 = Pvs_size3 / 8;$

% Bandwidth Video Streaming

$Ba = Pa_size * Bcodeca / PLa;$

$Bv1 = Pv_size1 * Bcodecv1 / PLv1;$

$Bv2 = Pv_size2 * Bcodecv2 / PLv2;$

$Bv3 = Pv_size3 * Bcodecv3 / PLv3;$

$Bvs1 = Ba + Bv1;$

$Bvs2 = Ba + Bv2;$

$Bvs3 = Ba + Bv3;$

$Bvs_kbps1 = Bvs1 / 1000;$

$Bvs_kbps2 = Bvs2 / 1000;$

$Bvs_kbps3 = Bvs3 / 1000;$

% PERHITUNGAN DELAY PROSES ENKAPSULASI DAN DEKAPSULASI

% server

$MTU = 1500;$

$header_eth = 14;$

$header_UDP = 8;$

$header_IP = 40;$

$FCS = 4;$

$C_server = 10^9;$

$N_frame_eth_server1 = Pvs_size_byte1 / MTU;$

$N_frame_eth_server2 = Pvs_size_byte2 / MTU;$

$N_frame_eth_server3 = Pvs_size_byte3 / MTU;$

$W_frame_server1 = Pvs_size_byte1 + (2 * (header_UDP + header_IP + header_eth + FCS));$

$W_frame_server2 = Pvs_size_byte2 + (2 * (header_UDP + header_IP + header_eth + FCS));$

$W_frame_server3 = Pvs_size_byte3 + (2 * (header_UDP + header_IP + header_eth + FCS));$

$te11 = (W_frame_server1 - Pvs_size_byte1) * 8 / C_server;$

$te12 = (W_frame_server2 - Pvs_size_byte2) * 8 / C_server;$

$te13 = (W_frame_server3 - Pvs_size_byte3) * 8 / C_server;$

% PDN-GW

$header_GTP = 8;$

$C_PDNGW = 10^8;$

$FCS = 4;$

$W_data_PDNGW1 = W_frame_server1 - (2 * (header_UDP + header_IP + header_eth + FCS));$

$W_data_PDNGW2 = W_frame_server2 - (2 * (header_UDP + header_IP + header_eth + FCS));$

$W_data_PDNGW3 = W_frame_server3 - (2 * (header_UDP + header_IP + header_eth + FCS));$

$td1 = (W_frame_server1 - W_data_PDNGW1) * 8 / C_PDNGW;$

$td2 = (W_frame_server2 - W_data_PDNGW2) * 8 / C_PDNGW;$

$td3 = (W_frame_server3 - W_data_PDNGW3) * 8 / C_PDNGW;$

```
MSS=MTU-header_GTP-header_UDP-header_IP;
N_dtgrm_PDNGW1=W_data_PDNGW1/MSS;
N_dtgrm_PDNGW2=W_data_PDNGW2/MSS;
N_dtgrm_PDNGW3=W_data_PDNGW3/MSS;
W_dtgrm_PDNGW1=W_data_PDNGW1+(2*(header_GTP+header_UDP+header_IP));
W_dtgrm_PDNGW2=W_data_PDNGW2+(2*(header_GTP+header_UDP+header_IP));
W_dtgrm_PDNGW3=W_data_PDNGW3+(2*(header_GTP+header_UDP+header_IP));
N_frame_eth_PDNGW1=W_dtgrm_PDNGW1/MTU;
N_frame_eth_PDNGW2=W_dtgrm_PDNGW2/MTU;
N_frame_eth_PDNGW3=W_dtgrm_PDNGW3/MTU;
W_frame_PDNGW1=W_dtgrm_PDNGW1+(2*(header_eth+FCS));
W_frame_PDNGW2=W_dtgrm_PDNGW2+(2*(header_eth+FCS));
W_frame_PDNGW3=W_dtgrm_PDNGW3+(2*(header_eth+FCS));
te21=(W_frame_PDNGW1-W_data_PDNGW1)*8/C_PDNGW;
te22=(W_frame_PDNGW2-W_data_PDNGW2)*8/C_PDNGW;
te23=(W_frame_PDNGW3-W_data_PDNGW3)*8/C_PDNGW;
```

```
%SGW
```

```
C_SGW=10^8;
```

```
FCS=4;
```

```
W_data_SGW1=W_frame_PDNGW1-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW2=W_frame_PDNGW2-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_SGW3=W_frame_PDNGW3-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
td21=(W_frame_PDNGW1-W_data_SGW1)*8/C_SGW;
```

```
td22=(W_frame_PDNGW2-W_data_SGW2)*8/C_SGW;
```

```
td23=(W_frame_PDNGW3-W_data_SGW3)*8/C_SGW;
```

```
N_dtgrm_SGW1=W_data_SGW1/MSS;
```

```
N_dtgrm_SGW2=W_data_SGW2/MSS;
```

```
N_dtgrm_SGW3=W_data_SGW3/MSS;
```

```
W_dtgrm_SGW1=W_data_SGW1+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW2=W_data_SGW2+(2*(header_GTP+header_UDP+header_IP));
```

```
W_dtgrm_SGW3=W_data_SGW3+(2*(header_GTP+header_UDP+header_IP));
```

```
N_frame_eth_SGW1=W_dtgrm_SGW1/MTU;
```

```
N_frame_eth_SGW2=W_dtgrm_SGW2/MTU;
```

```
N_frame_eth_SGW3=W_dtgrm_SGW3/MTU;
```

```
W_frame_SGW1=W_dtgrm_SGW1+(2*(header_eth+FCS));
```

```
W_frame_SGW2=W_dtgrm_SGW2+(2*(header_eth+FCS));
```

```
W_frame_SGW3=W_dtgrm_SGW3+(2*(header_eth+FCS));
```

```
te31=(W_frame_SGW1-W_data_SGW1)*8/C_SGW;
```

```
te32=(W_frame_SGW2-W_data_SGW2)*8/C_SGW;
```

```
te33=(W_frame_SGW3-W_data_SGW3)*8/C_SGW;
```

```
%eNB
```

```
C_eNB=155.52*(10^6);
```

```
header_PDCP=1;
```

```
header_MAC=0.5;
```

```
header_RLC=2;
```

```
W_data_eNB1=W_frame_SGW1-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```
W_data_eNB2=W_frame_SGW2-
```

```
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
```

```

W_data_eNB3=W_frame_SGW3-
(2*(header_GTP+header_UDP+header_IP+header_eth+FCS));
td31=(W_frame_SGW1-W_data_eNB1)*8/C_eNB;
td32=(W_frame_SGW2-W_data_eNB2)*8/C_eNB;
td33=(W_frame_SGW3-W_data_eNB3)*8/C_eNB;
N_dtgrm_eNB1=W_data_eNB1/MTU;
N_dtgrm_eNB2=W_data_eNB2/MTU;
N_dtgrm_eNB3=W_data_eNB3/MTU;
W_frame_PDCP1=W_data_eNB1+header_PDCP;
W_frame_PDCP2=W_data_eNB2+header_PDCP;
W_frame_PDCP3=W_data_eNB3+header_PDCP;
N_frame_RLC1=W_frame_PDCP1/40;
N_frame_RLC2=W_frame_PDCP2/40;
N_frame_RLC3=W_frame_PDCP3/40;
W_frame_RLCtot1=31*42;
W_frame_RLCtot2=47*42;
W_frame_RLCtot3=63*42;
N_frame_MAC1=W_frame_RLCtot1/(40+header_RLC);
N_frame_MAC2=W_frame_RLCtot2/(40+header_RLC);
N_frame_MAC3=W_frame_RLCtot3/(40+header_RLC);
W_frame_MAC=42+header_MAC;
W_frame_eNB1=W_frame_MAC*31;
W_frame_eNB2=W_frame_MAC*47;
W_frame_eNB3=W_frame_MAC*63;
te41=(W_frame_eNB1-W_data_eNB1)*8/C_eNB;
te42=(W_frame_eNB2-W_data_eNB2)*8/C_eNB;
te43=(W_frame_eNB3-W_data_eNB3)*8/C_eNB;

%UE
C_UE=3.6*(10^6);
W_data_UE1=W_frame_eNB1-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
W_data_UE2=W_frame_eNB2-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
W_data_UE3=W_frame_eNB3-(6*header_GTP)-(8*header_UDP)-(8*header_IP)-
(2*header_PDCP)-(42*header_RLC)-(42*header_MAC);
td41=(W_frame_eNB1-W_data_UE1)*8/C_UE;
td42=(W_frame_eNB2-W_data_UE2)*8/C_UE;
td43=(W_frame_eNB3-W_data_UE3)*8/C_UE;

te_tot1=te11+te21+te31+te41;
te_tot2=te12+te22+te32+te42;
te_tot3=te13+te23+te33+te43;
td_tot1=td1+td21+td31+td41;
td_tot2=td2+td22+td32+td42;
td_tot3=td3+td23+td33+td43;

```

%PERHITUNGAN DELAY TRANSMISI

```

cr=3.84*(10^6);
rf1=0.01;
tt11=W_frame_server1*8/C_server;
tt12=W_frame_server2*8/C_server;
tt13=W_frame_server3*8/C_server;
tt21=W_frame_PDNGW1*8/C_PDNGW;

```

```

tt22=W_frame_PDNGW2*8/C_PDNGW;
tt23=W_frame_PDNGW3*8/C_PDNGW;
tt31=W_frame_SGW1*8/C_SGW;
tt32=W_frame_SGW2*8/C_SGW;
tt33=W_frame_SGW3*8/C_SGW;
tt41=(W_frame_eNB1*8)/((cr*rf1/2560)*C_eNB);
tt42=(W_frame_eNB2*8)/((cr*rf1/2560)*C_eNB);
tt43=(W_frame_eNB3*8)/((cr*rf1/2560)*C_eNB);
tt_tot1=tt11+tt21+tt31+tt41;
tt_tot2=tt12+tt22+tt32+tt42;
tt_tot3=tt13+tt23+tt33+tt43;

```

%PERHITUNGAN DELAY PROPAGASI

```

d_server_PDNGW=5000;
d_PDNGW_SGW=5000;
d_SGW_eNB=5000;
d_eNB_UE1=1000;

```

```

c=3*10^6;
tp1=2*d_server_PDNGW/c;
tp2=2*d_PDNGW_SGW/c;
tp3=2*d_SGW_eNB/c;
tp41=31*d_eNB_UE1/c;
tp42=47*d_eNB_UE1/c;
tp43=63*d_eNB_UE1/c;
tp_tot1=tp1+tp2+tp3+tp41;
tp_tot2=tp1+tp2+tp3+tp42;
tp_tot3=tp1+tp2+tp3+tp43;

```

%PERHITUNGAN DELAY ANTRIAN

```

miu_PDNGW1=C_PDNGW/(W_frame_PDNGW1*8);
miu_PDNGW2=C_PDNGW/(W_frame_PDNGW2*8);
miu_PDNGW3=C_PDNGW/(W_frame_PDNGW3*8);
miu_SGW1=C_SGW/(W_frame_SGW1*8);
miu_SGW2=C_SGW/(W_frame_SGW2*8);
miu_SGW3=C_SGW/(W_frame_SGW3*8);
miu_eNB1=C_eNB/(W_frame_eNB1*8);
miu_eNB2=C_eNB/(W_frame_eNB2*8);
miu_eNB3=C_eNB/(W_frame_eNB3*8);
ro0=0.1;
ro1=0.9;
ro=[ro0:0.1:ro1];
lamda_PDNGW1=miu_PDNGW1.*ro;
lamda_PDNGW2=miu_PDNGW2.*ro;
lamda_PDNGW3=miu_PDNGW3.*ro;
lamda_SGW1=miu_SGW1.*ro;
lamda_SGW2=miu_SGW2.*ro;
lamda_SGW3=miu_SGW3.*ro;
lamda_eNB1=miu_eNB1.*ro;
lamda_eNB2=miu_eNB2.*ro;
lamda_eNB3=miu_eNB3.*ro;
tw11=lamda_PDNGW1./(miu_PDNGW1*(miu_PDNGW1-
lamda_PDNGW1))+1/miu_PDNGW1);

```

```

tw12=lamda_PDNGW2./(miu_PDNGW2*(miu_PDNGW2-
lamda_PDNGW2))+(1/miu_PDNGW2);
tw13=lamda_PDNGW3./(miu_PDNGW3*(miu_PDNGW3-
lamda_PDNGW3))+(1/miu_PDNGW3);
tw21=lamda_SGW1./(miu_SGW1*(miu_SGW1-lamda_SGW1))+(1/miu_SGW1);
tw22=lamda_SGW2./(miu_SGW2*(miu_SGW2-lamda_SGW2))+(1/miu_SGW2);
tw23=lamda_SGW3./(miu_SGW3*(miu_SGW3-lamda_SGW3))+(1/miu_SGW3);
tw31=lamda_eNB1./(miu_eNB1*(miu_eNB1-lamda_eNB1))+(1/miu_eNB1);
tw32=lamda_eNB2./(miu_eNB2*(miu_eNB2-lamda_eNB2))+(1/miu_eNB2);
tw33=lamda_eNB3./(miu_eNB3*(miu_eNB3-lamda_eNB3))+(1/miu_eNB3);
tw_tot1=tw11+tw21+tw31;
tw_tot2=tw12+tw22+tw32;
tw_tot3=tw13+tw23+tw33;

```

%PERHITUNGAN DELAY CODEC

```

ta=0.04;
tv=0.3;
t_CODEC=ta+tv;

```

%PERHITUNGAN DELAY TOTAL

```

t_tot1=t_CODEC+te_tot1+td_tot1+tt_tot1+tp_tot1+tw_tot1;
t_tot2=t_CODEC+te_tot2+td_tot2+tt_tot2+tp_tot2+tw_tot1;
t_tot3=t_CODEC+te_tot3+td_tot3+tt_tot3+tp_tot3+tw_tot3;

```

%PERHITUNGAN TIME OUT INTERVAL

```

header_frame=8;
tl1=((W_frame_eNB1+header_frame)*8)/C_UE;
tl2=((W_frame_eNB2+header_frame)*8)/C_UE;
tl3=((W_frame_eNB3+header_frame)*8)/C_UE;
tp1=tp_tot1;
tp2=tp_tot2;
tp3=tp_tot3;
t_pro1=te_tot1+td_tot1+tw_tot1;
t_pro2=te_tot2+td_tot2+tw_tot2;
t_pro3=te_tot3+td_tot3+tw_tot3;
t_out1=(2*tp_tot1)+(2*tl1)+t_pro1;
t_out2=(2*tp_tot2)+(2*tl2)+t_pro2;
t_out3=(2*tp_tot3)+(2*tl3)+t_pro3;

```

%PERHITUNGAN KONSTANTA

```

a1=1+(t_out1/tl1);
a2=1+(t_out2/tl2);
a3=1+(t_out3/tl3);

```

% PERHITUNGAN PROBABILITAS PACKET LOSS

```

%packet loss server
Pb=10^-8;
p_VS1=Pvs_size1*pb;
p_VS2=Pvs_size2*pb;
p_VS3=Pvs_size3*pb;

```

%BIT ERROR RATE

PG=12;

NF=7;

Pt=46;

Lt=2;

Lr=0;

Gt=18;

Gr=0;

K=1.381*10⁻²³;

T=300;

B=10*10⁶;

R=3.6*10⁶;

c=3*10⁸;

f=2.310⁹;

d=1000;

M=64;

No=10*log10(K*T)+10*log10(B)+NF;

lamda=c/f;

FSL=20*log10((4*pi*d)/lamda);

Pr=Pt-FSL-Lt-Lr-Gr-Gt;

SNR=Pr-No;

Eb_per_No=SNR-10*log10(B/R);

x=sqrt((3*Eb_per_No*log2(M))/(M-1));

BER=((6*(sqrt(M)-1))/(sqrt(M))*log2(M))*(0.5*erfc(x/sqrt(2)));

%packet loss LTE

p_LTE1=(W_frame_eNB1*8)*BER;

p_LTE2=(W_frame_eNB2*8)*BER;

p_LTE3=(W_frame_eNB3*8)*BER;

%total packet loss

p_tot1=1-((1-p_LTE1)*(1-p_VS1));

p_tot2=1-((1-p_LTE2)*(1-p_VS2));

p_tot3=1-((1-p_LTE3)*(1-p_VS3));

%PERHITUNGAN THROUGHPUT

T1=(1-p_tot1)/(t1*(1+(a1-1).*p_tot1));

T2=(1-p_tot2)/(t2*(1+(a2-1).*p_tot2));

T3=(1-p_tot3)/(t3*(1+(a3-1).*p_tot3));

T1_kbps=T1*W_frame_eNB1*8/1000;

T2_kbps=T2*W_frame_eNB2*8/1000;

T3_kbps=T3*W_frame_eNB3*8/1000;

%GRAFIK 128,256,384

figure

plot(ro,T1_kbps,'-^r',ro,T2_kbps,'-*g',ro,T3_kbps,'-^b')

grid on

title('Grafik Throughput Terhadap Faktor Utilisasi');

legend('Bit Rate=128 kbps','Bit Rate=256 kbps','Bit Rate=384 kbps')

xlabel ('Faktor Utilisasi')

ylabel ('Throughput (kbps)')

LAMPIRAN 5

PERCOBAAN PERFORMANSI VIDEO STREAMING OVER LAN (LOCAL AREA NETWORK)

1.1 Tujuan Percobaan

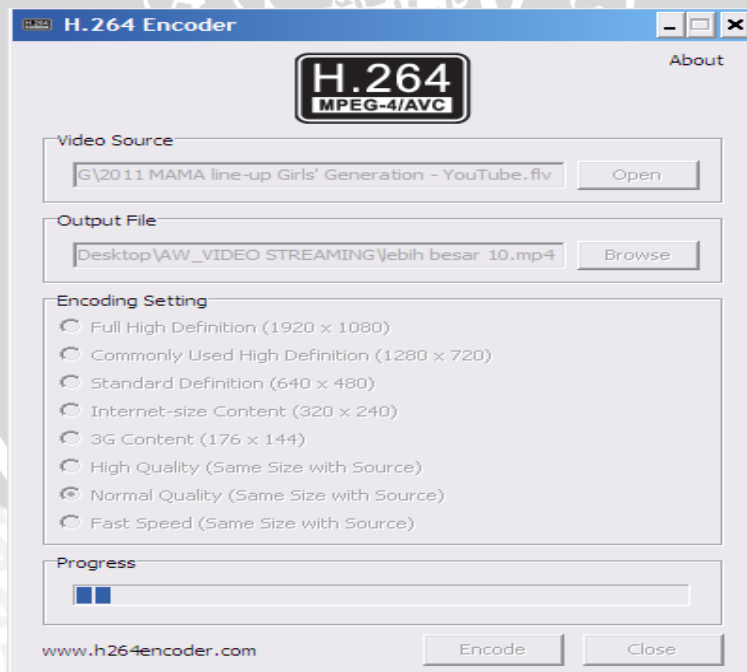
- Mengetahui performansi video streaming yang meliputi *bandwidth*, *delay*, *packet loss*, *throughput*.

1.2 Peralatan yang Diperlukan

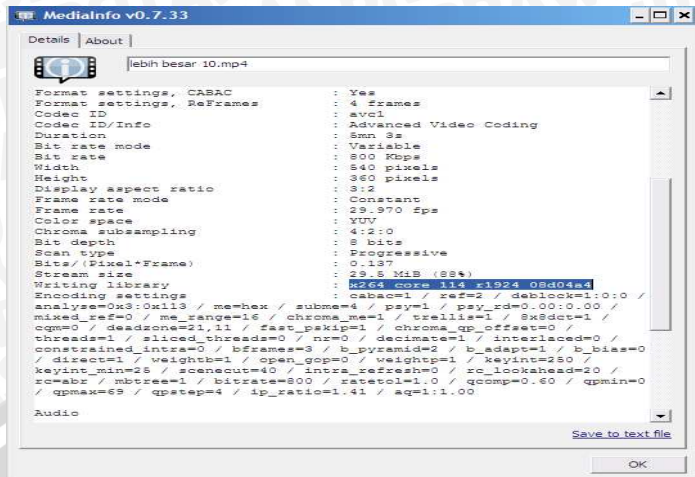
- 2 unit CPU sebagai Server dan Client
- Software H.264 Encoder
- Software VLC Media Player
- Software Wireshark

1.3 Prosedur Percobaan**I. Convert File**

1. Buka aplikasi H.264 Encoder
2. Pilih normal quality
3. Encode

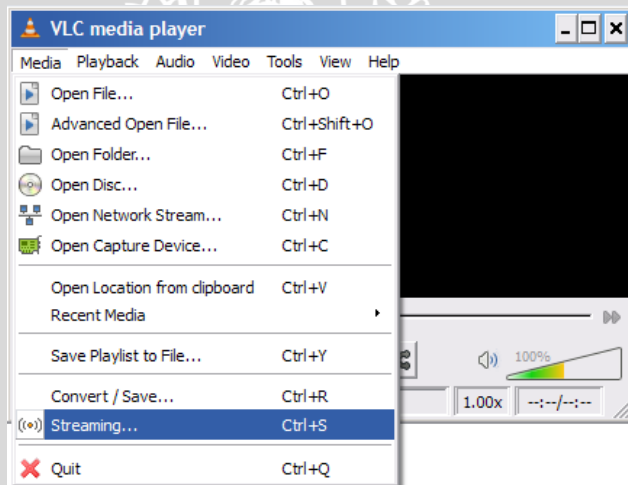


4. Cek file (buka file → media info)

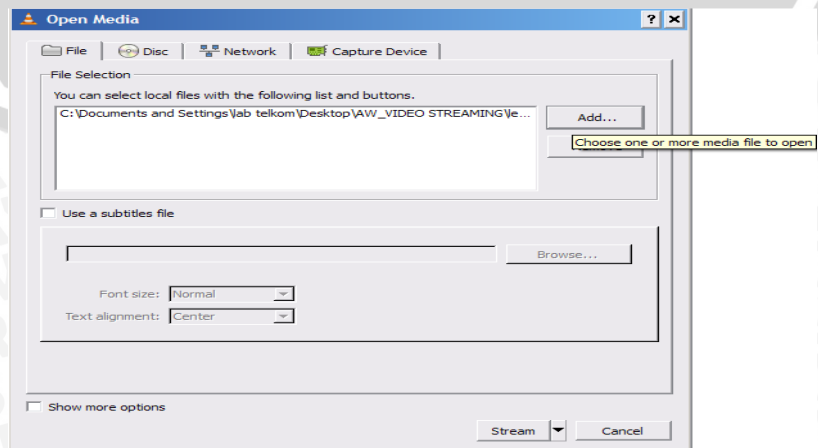


II. Streaming Video (Server)

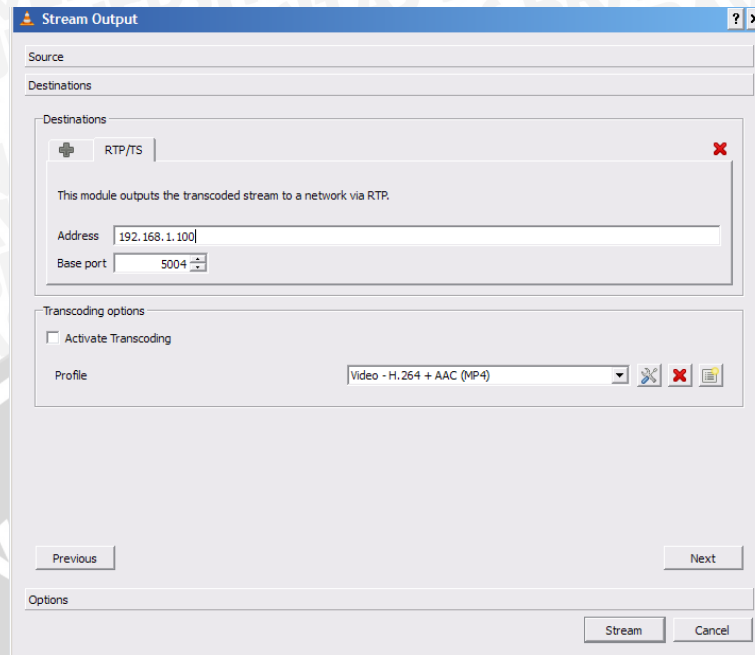
1. Buka aplikasi VLC Player
2. Klik Media → streaming



3. Klik add → pilih file yang sudah diconvert
4. Pilih mode stream



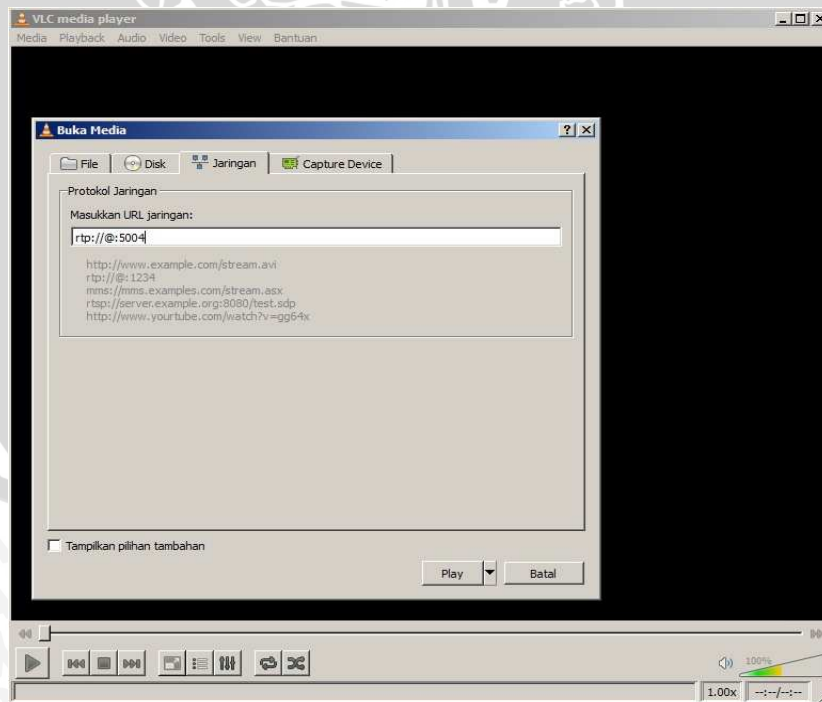
- Pilih RTP (MPEG Transport Stream) di new destination → klik add → masukkan IP client → klik stream



- Tunggu hingga file selesai distreaming.

III. Streaming Video (Client)

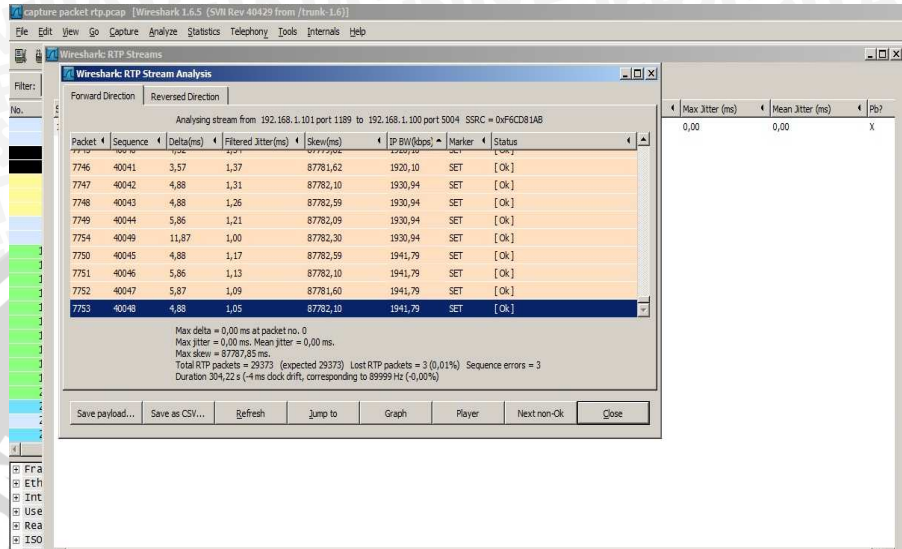
- Buka aplikasi VLC Player sebelum file distreaming dari server
- Klik Media → Buka stream jaringan (Ctrl+N) → masukkan URL jaringan → play



- Ketika file sudah distreaming dari server, maka otomatis client akan segera memainkan file yang dikirim.

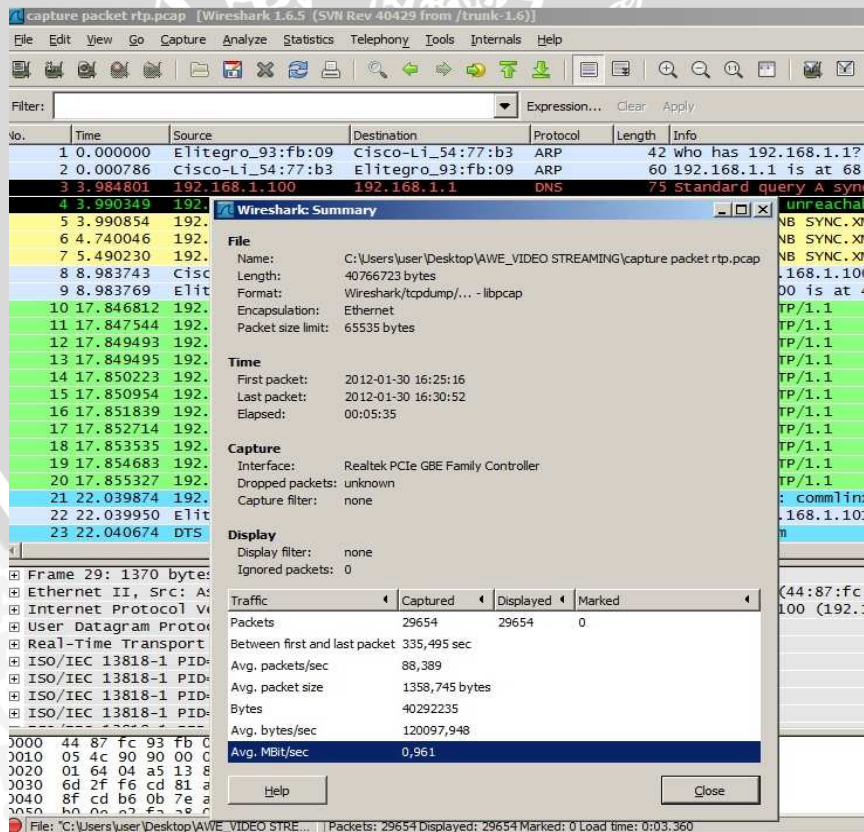
IV. Performansi Video Streaming

1. Buka aplikasi wireshark sebelum file distreaming dari server → client
2. Setelah file selesai distreaming ke client, simpan/cuplik dengan wireshark
3. Perhitungan bandwidth



Klik telephony → RTP → show all stream → analyze

4. Perhitungan delay dan throughput



Klik statistic → summary

5. Perhitungan packet loss

The screenshot shows the 'Wireshark RTP Streams' window. It displays a table of detected RTP streams. The first stream is selected, showing the following details:

No.	Src IP addr	Src port	Dst IP addr	Dst port	SSRC	Payload	Packets	Lost	Max Delta (ms)	Max Jitter (ms)	Mean Jitter (ms)	Pk7
1	192.168.1.101	1189	192.168.1.100	5004	0xF6CD814B	MPEG-II streams	29370	3 (0,0%)	0,00	0,00	0,00	X

Below the table, there are instructions: 'Select a forward stream with left mouse button, and then Select a reverse stream with Ctrl + left mouse button'. At the bottom, there are buttons: 'Unselect', 'Find Reverse', 'Save As', 'Mark Packets', 'Prepare Filter', 'Copy', 'Analyze', and 'Close'.

Klik telephony → RTP → show all stream

