

BAB IV HASIL DAN ANALISA

Penelitian ini dilaksanakan untuk menganalisa pengaruh algoritma penjadwalan yang berbeda pada kelas rtPS. Algoritma penjadwalan yang digunakan pada kelas rtPS adalah WRR dan TRS_RR. Simulasi ini dilakukan dengan membangkitkan beberapa jenis trafik, yaitu cbr pada UGS, vbr pada rtPS dan FTP pada BE. Pengaruh penjadwalan akan diamati dengan melihat nilai *throughput* dan *jitter* pada masing-masing penjadwalan.

4.1. Parameter Utama dalam Simulasi

Untuk melaksanakan simulasi ini, beberapa parameter telah ditentukan dalam script sebagai berikut :

Tabel 4.1. Parameter Utama dalam Simulasi

Characteristic	Values
Frame duration Mac/802_16 set frame_duration_ 0.020	20 ms
Propagation Model set opt(prop) Propagation/TwoRayGround ;# radio- propagation model	Two Ray Ground
Antenna Model set opt(ant) Antenna/OmniAntenna ;# antenna model	Omni antenna
Transmit power Phy/WirelessPhy set Pt_ 0.025 ;#transmit power	0.025
Receive power threshold Phy/WirelessPhy set RXThresh_ 2.025e-12 ;#500m radius	$2025e^{-12}$
Carrier sense power threshold Phy/WirelessPhy set CStresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]	0.9*Receive power threshold

Simulasi diset, sehingga trafik dimulai pada detik ke 7 dan berakhir pada detik ke 80.

4.2. Menjalankan Simulasi

Pada simulasi ini, UGS diset mendapat prioritas utama, kemudian algoritma penjadwalan WRR dan TRS_RR akan diterapkan pada rtPS, sedangkan BE menggunakan algoritma penjadwalan sederhana yaitu RR. Simulasi dijalankan dengan skenario sebagai berikut :

1. rtPS menggunakan algoritma penjadwalan WRR :
 - a. simulasi memakai modulasi QPSK $\frac{3}{4}$,
 - b. simulasi memakai modulasi 16 QAM $\frac{3}{4}$, dan
 - c. simulasi memakai modulasi 64 QAM $\frac{3}{4}$.
2. rtPS menggunakan algoritma penjadwalan TRS_RR :
 - a. simulasi memakai modulasi QPSK $\frac{3}{4}$,
 - b. simulasi memakai modulasi 16 QAM $\frac{3}{4}$, dan
 - c. simulasi memakai modulasi 64 QAM $\frac{3}{4}$.

Selanjutnya *throughput* dan *jitter* yang dihasilkan dari masing-masing skenario akan diperbandingkan.

Untuk menjalankan simulasi dengan skenario tersebut diatas, dilaksanakan dengan mengetikkan command sebagai berikut :

Skenario 1 (WRR – RR) :

a. QPSK $\frac{3}{4}$:

- Command:
ns datarate.tcl OFDM_QPSK_3_4 0.03125 WRR RR
- *Throughput*
perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > WRR1-Throughput.tr
- *Jitter*
awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > WRR1-Jitter1.tr
awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > WRR1-Jitter2.tr
awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > WRR1-Jitter3.tr
awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > WRR1-Jitter4.tr

b. 16 QAM $\frac{3}{4}$:

- Command:
ns datarate.tcl OFDM_16QAM_3_4 0.03125 WRR RR
- *Throughput*
perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > WRR2-Throughput.tr

- *Jitter*

```
awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > WRR2-Jitter1.tr
awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > WRR2-Jitter2.tr
awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > WRR2-Jitter3.tr
awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > WRR2-Jitter4.tr
```

c. 64 QAM $\frac{3}{4}$:

- *Command:*

```
ns datarate.tcl OFDM_64QAM_3_4 0.03125 WRR RR
```

- *Throughput*

```
perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > WRR3-Throughput.tr
```

- *Jitter*

```
awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > WRR3-Jitter1.tr
awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > WRR3-Jitter2.tr
awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > WRR3-Jitter3.tr
awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > WRR3-Jitter4.tr
```

Skenario 2 (TRS_RR – RR) :

a. QPSK $\frac{3}{4}$:

- *Command:*

```
ns datarate.tcl OFDM_QPSK_3_4 0.03125 TRS_RR RR
```

- *Throughput*

```
perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > TRS_RR1-Throughput.tr
```

- *Jitter*

```
awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR1-Jitter1.tr
awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR1-Jitter2.tr
awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR1-Jitter3.tr
awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR1-Jitter4.tr
```

b. 16QAM $\frac{3}{4}$:

- *Command:*

```
ns datarate.tcl OFDM_16QAM_3_4 0.03125 TRS_RR RR
```

- *Throughput*

```
perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > TRS_RR2-Throughput.tr
```

- *Jitter*

```
awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR2-Jitter1.tr
awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR2-Jitter2.tr
awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR2-Jitter3.tr
awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR2-Jitter4.tr
```

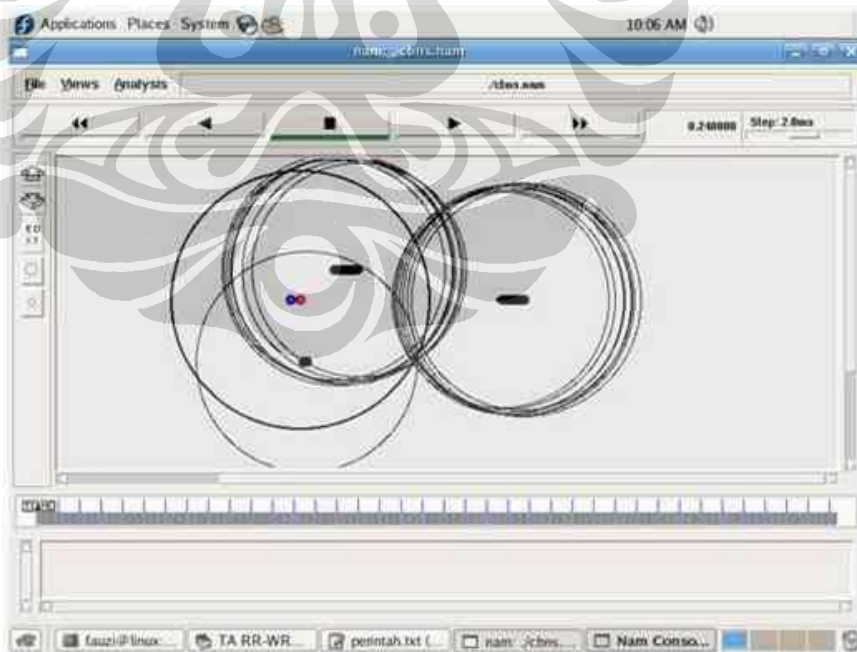
b. 64QAM $\frac{3}{4}$:

- **Command:**
`ns datarate.tcl OFDM_64QAM_3_4 0.03125 TRS_RR RR`
- **Throughput**
`perl datarateThroughput.pl datarate.tr 0.0 0 1 1 1000 > TRS_RR3-Throughput.tr`
- **Jitter**
`awk -f instantJitter1.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR3-Jitter1.tr`
`awk -f instantJitter2.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR3-Jitter2.tr`
`awk -f instantJitter3.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR3-Jitter3.tr`
`awk -f instantJitter4.awk flow=1 src=0 pkt=1000 datarate.tr > TRS_RR3-Jitter4.tr`

4.3. Perbandingan dan Analisa Hasil Simulasi

4.3.1. NAM

Dari eksekusi command pada setiap skenario untuk menjalankan network simulator melalui perintah “ns” terhadap script WiMAX yang dibuat (file `datarate.tcl`) tersebut diatas, maka akan dibangun suatu topologi berdasarkan script yang dibuat yang hasilnya akan ditunjukkan melalui Network Animator (NAM). Output dari NAM tersebut dapat dilihat pada Gambar 4.1. di bawah ini.



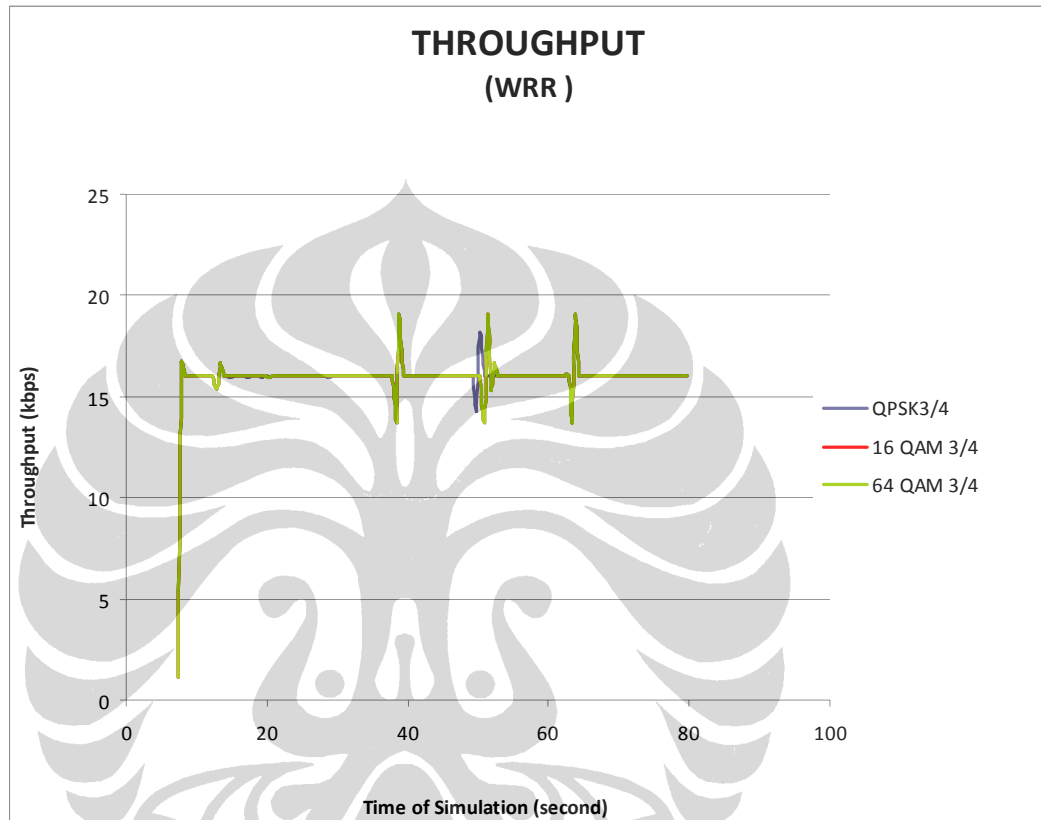
Gambar 4.1. Gambar Animasi dari Topologi yang berhasil dibangun

4.3.2. Throughput

Berdasarkan perintah untuk mendapatkan *throughput* pada setiap skenario, maka diperoleh grafik *Throughput* sebagai berikut :

a. Pada skenario pertama :

- WRR



Gambar 4.2. Grafik Perbandingan *Throughput* WRR pada berbagai Modulasi

Hasil dari simulasi dengan algoritma penjadwalan WRR yang dilakukan pada modulasi QPSK $\frac{3}{4}$, 16 QAM $\frac{3}{4}$ dan 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.2. Berdasarkan gambar grafik *throughput* tersebut diatas, diperoleh nilai rata-rata *throughput* sebagai berikut :

WRR (QPSK $\frac{3}{4}$) : 15,9174707 kbps

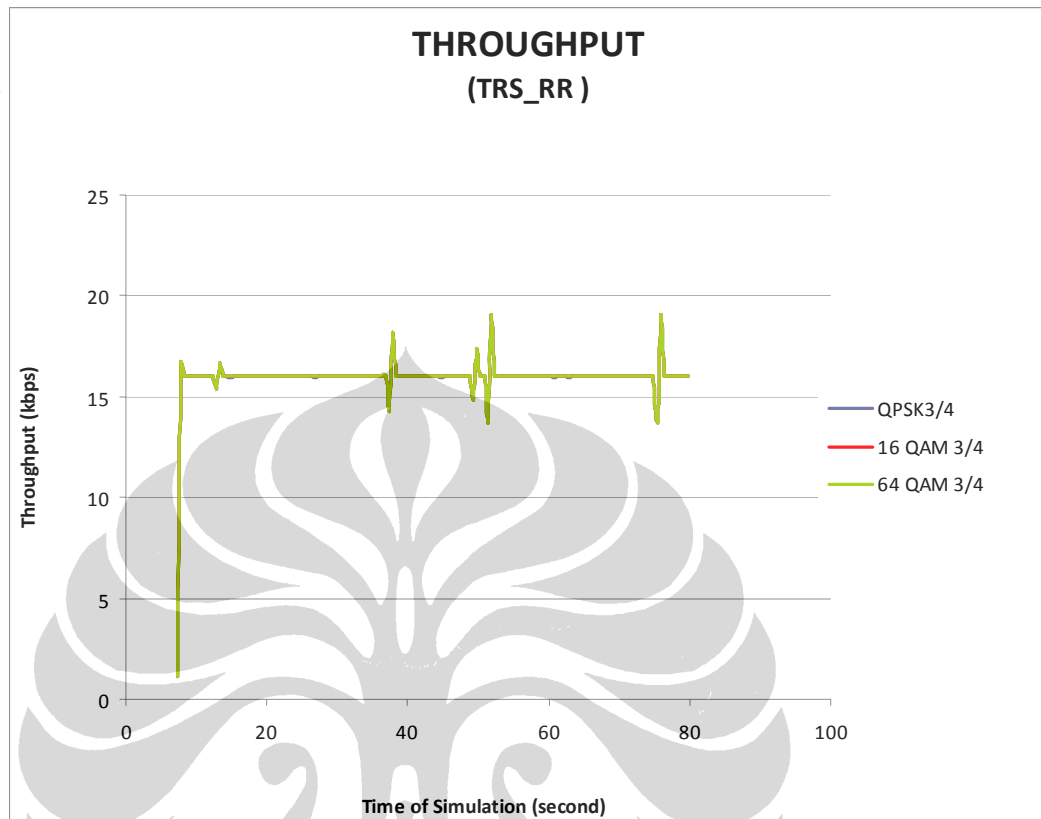
WRR (16QAM $\frac{3}{4}$) : 15,9203679 kbps

WRR (64QAM $\frac{3}{4}$) : 15,9203659 kbps

Berdasarkan nilai tersebut, terlihat bahwa dengan semakin tinggi modulasi yang digunakan, rata-rata *throughput* juga cenderung semakin besar. Rata-rata *throughput* tertinggi pada algoritma penjadwalan WRR adalah pada saat menggunakan modulasi 16 QAM $\frac{3}{4}$, yaitu sebesar 15,9203679 kbps.

b. Pada skenario kedua :

- TRS_RR :



Gambar 4.3. Grafik Perbandingan *Throughput* TRS_RR pada berbagai Modulasi

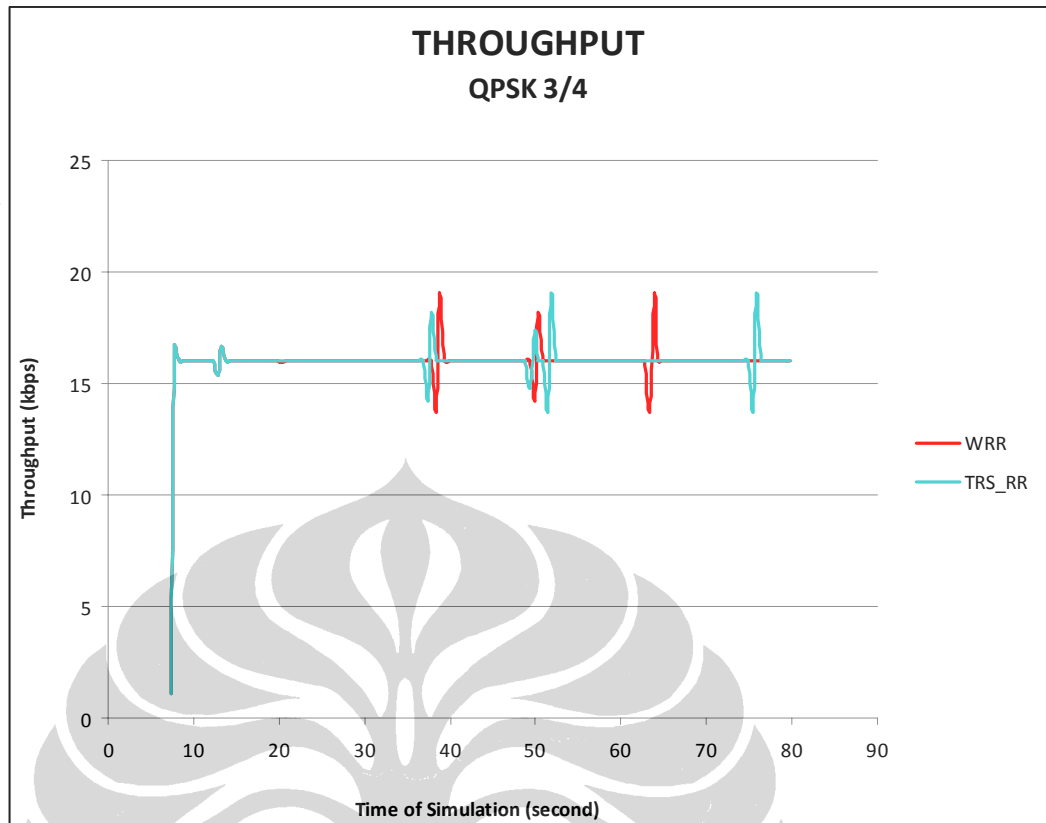
Hasil dari simulasi dengan algoritma penjadwalan TRS_RR yang dilakukan pada modulasi QPSK $\frac{3}{4}$, 16 QAM $\frac{3}{4}$ dan 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.3. Berdasarkan gambar grafik *throughput* tersebut diatas, diperoleh nilai rata-rata *throughput* sebagai berikut :

TRS_RR (QPSK $\frac{3}{4}$)	:	15,918944 kbps
TRS_RR (16QAM $\frac{3}{4}$)	:	15,9189905 kbps
TRS_RR (64QAM $\frac{3}{4}$)	:	15,919 kbps

Berdasarkan nilai tersebut, terlihat bahwa dengan semakin tinggi modulasi yang digunakan, rata-rata *throughput* juga cenderung semakin besar. Dimana rata-rata *throughput* tertinggi pada algoritma penjadwalan TRS_RR adalah pada saat menggunakan modulasi 64 QAM $\frac{3}{4}$, yaitu sebesar 15,919 kbps.

c. Jika *throughput* WRR dan TRS_RR dibandingkan pada modulasi yang sama :

- Modulasi QPSK $\frac{3}{4}$:



Gambar 4.4. Grafik Perbandingan *Throughput* WRR dan TRS_RR pada Modulasi QPSK 3/4

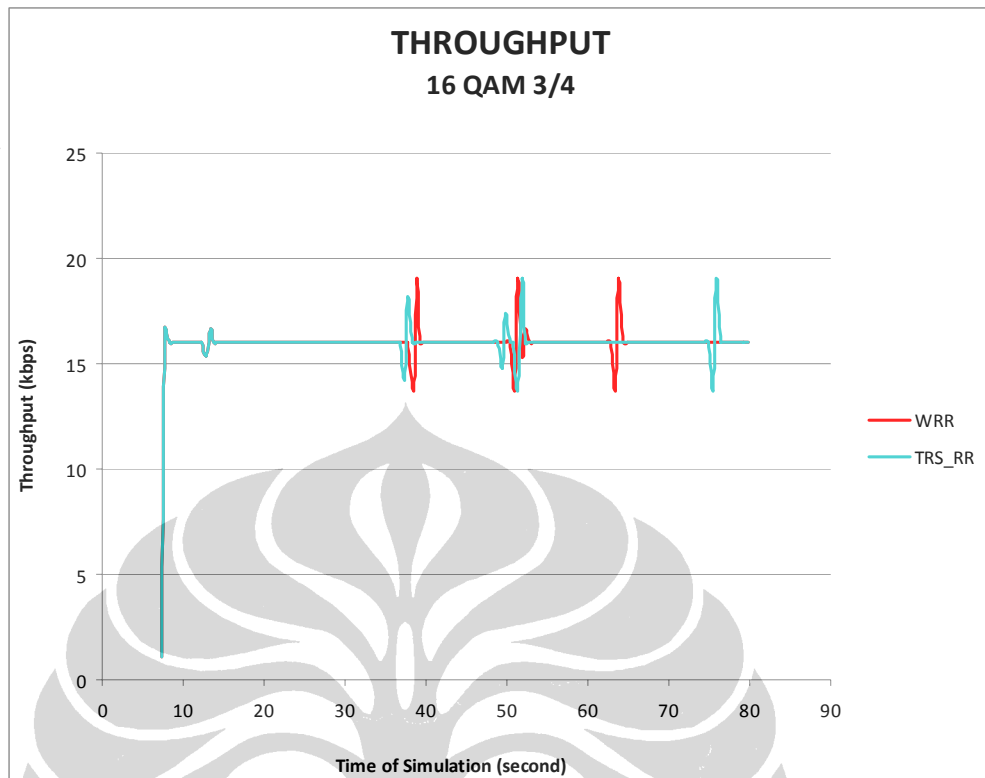
Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi QPSK $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.4. Berdasarkan gambar grafik *throughput* tersebut diatas, diperoleh nilai rata-rata *throughput* sebagai berikut :

WRR (QPSK3/4) : 15,9174707 kbps

TRS_RR (QPSK3/4) : 15,918944 kbps

Berdasarkan nilai tersebut, terlihat bahwa pada modulasi QPSK $\frac{3}{4}$ TRS_RR memiliki rata-rata *throughput* lebih tinggi daripada WRR, yaitu sebesar 15,918944 kbps.

- Modulasi 16 QAM $\frac{3}{4}$:



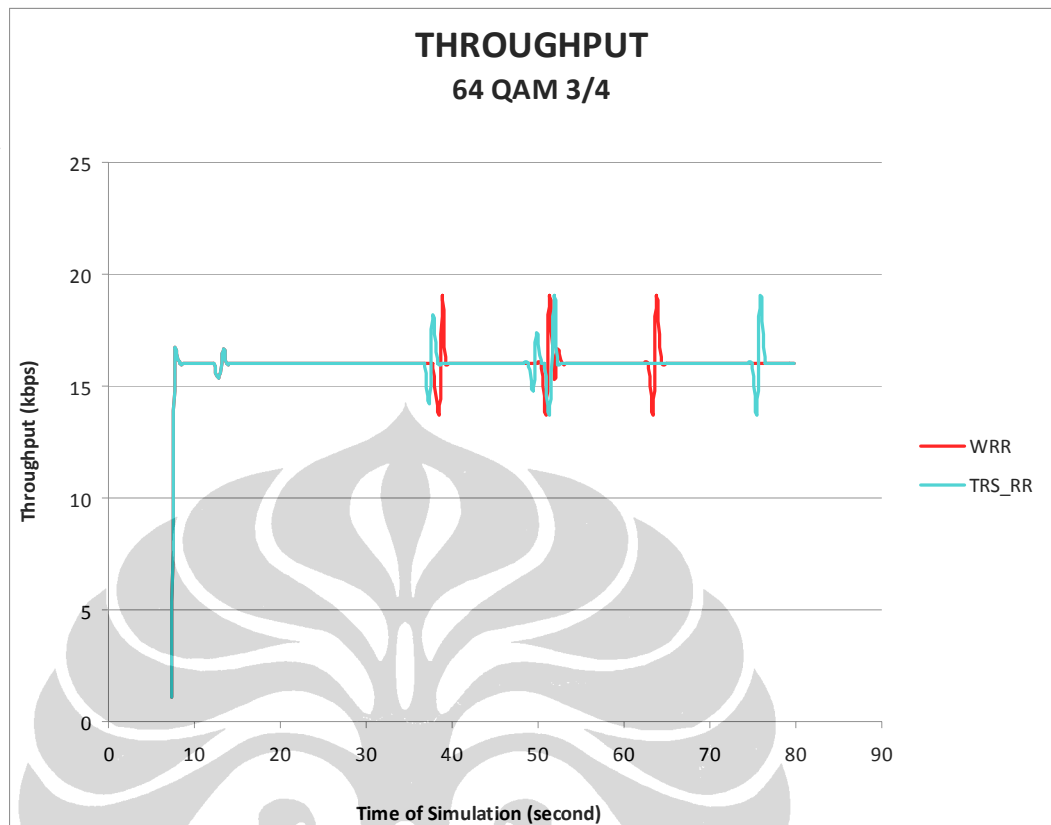
Gambar 4.5. Grafik Perbandingan *Throughput* WRR dan TRS_RR pada Modulasi 16 QAM $\frac{3}{4}$

Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi 16 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.5. Berdasarkan gambar grafik *throughput* tersebut diatas, diperoleh nilai rata-rata *throughput* sebagai berikut :

WRR (16QAM $\frac{3}{4}$) : 15,9203679 kbps.
 TRS_RR (16QAM $\frac{3}{4}$) : 15,918905 kbps.

Berdasarkan nilai tersebut, terlihat bahwa pada modulasi 16 QAM $\frac{3}{4}$ WRR memiliki rata-rata *throughput* lebih tinggi daripada TRS_RR, yaitu sebesar 15,9203679 kbps.

- Modulasi 64 QAM $\frac{3}{4}$:



Gambar 4.6. Grafik Perbandingan *Throughput* WRR dan TRS_RR pada Modulasi 64 QAM $\frac{3}{4}$

Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.6. Berdasarkan gambar grafik *throughput* tersebut diatas, diperoleh nilai rata-rata *throughput* sebagai berikut :

WRR (64QAM $\frac{3}{4}$) : 15,9203659 kbps.

TRS_RR (64QAM $\frac{3}{4}$) : 15,919 kbps.

Berdasarkan nilai tersebut, terlihat bahwa pada modulasi 64 QAM $\frac{3}{4}$ WRR memiliki rata-rata *throughput* lebih tinggi daripada TRS_RR, yaitu sebesar 15,9203659 kbps.

Berdasarkan pengamatan terhadap keseluruhan grafik *throughput* tersebut diatas, dapat dilihat bahwa :

Berdasarkan nilai rata-rata *throughput*, terlihat bahwa :

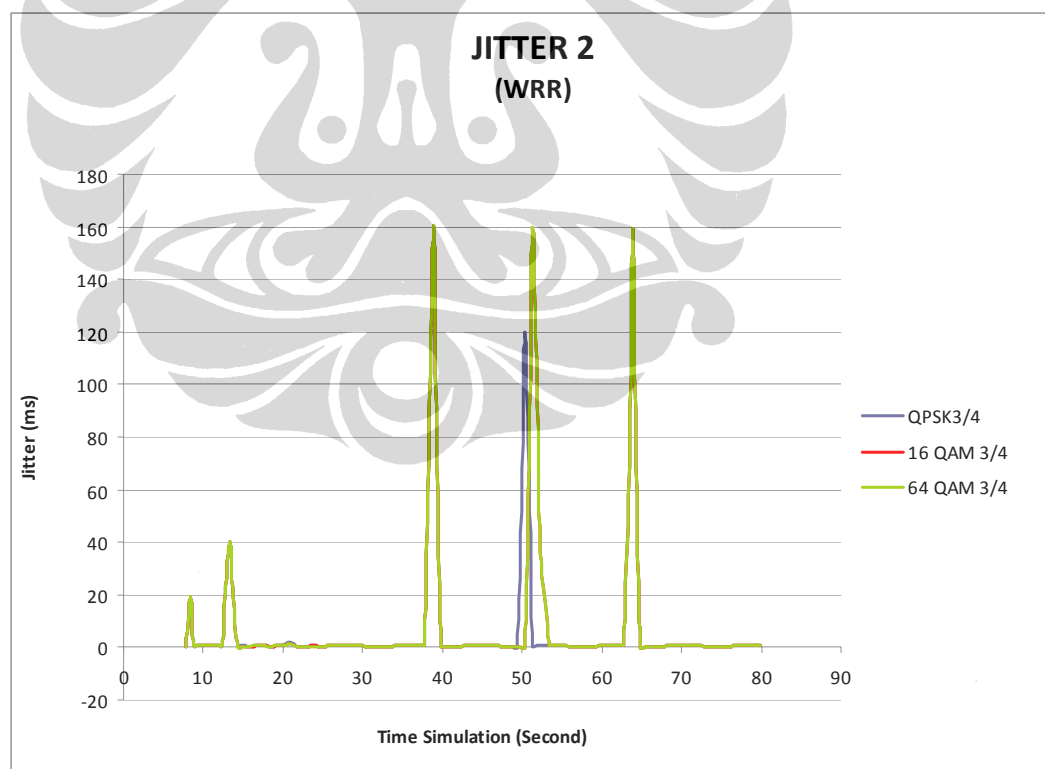
- Semakin tinggi modulasi yang digunakan, nilai rata-rata *throughput* pada algoritma penjadwalan WRR maupun TRS-RR cenderung semakin besar;
- algoritma penjadwalan WRR memiliki nilai *throughput* tertinggi pada modulasi 16 QAM $\frac{3}{4}$ yaitu sebesar 15,9203679 kbps;
- algoritma penjadwalan TRS-RR memiliki nilai *throughput* tertinggi pada modulasi 64 QAM $\frac{3}{4}$ yaitu sebesar 15,919 kbps;
- Namun, terlihat bahwa semakin tinggi modulasi yang digunakan, ternyata rata-rata *throughput* WRR lebih baik dibandingkan TRS-RR.

4.3.3. Jitter

Untuk menganalisa nilai jitter, diambil jitter 2 sebagai sampel. Berdasarkan perintah untuk mendapatkan *jitter* pada setiap skenario, maka diperoleh grafik *Jitter* sebagai berikut :

a. Pada skenario pertama :

- WRR



Gambar 4.7. Grafik Perbandingan *Jitter* WRR pada berbagai Modulasi

Hasil dari simulasi dengan algoritma penjadwalan WRR yang dilakukan pada modulasi QPSK $\frac{3}{4}$, 16 QAM $\frac{3}{4}$ dan 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana

terlihat pada Gambar 4.7. Berdasarkan gambar grafik *jitter* tersebut diatas, diperoleh nilai rata-rata *jitter* sebagai berikut :

WRR (QPSK $\frac{3}{4}$) : 7,215041 ms.

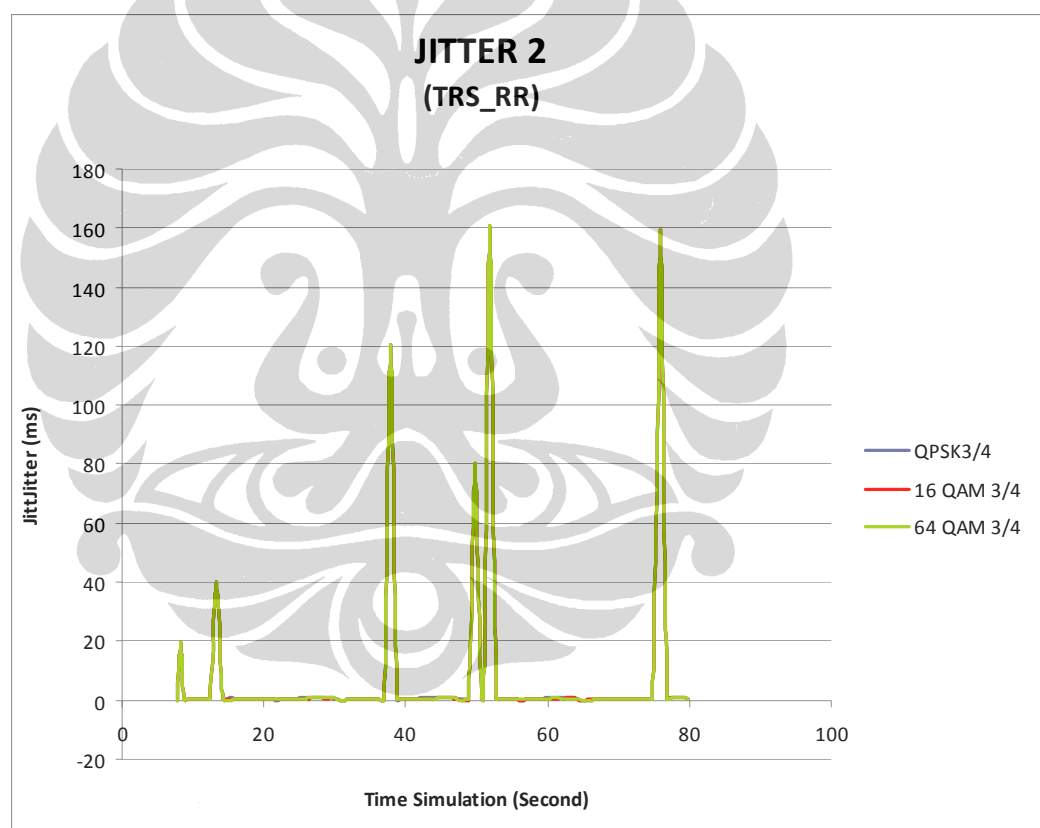
WRR (16QAM $\frac{3}{4}$) : 8,269179 ms.

WRR (64QAM $\frac{3}{4}$) : 8,266724 ms.

Berdasarkan nilai tersebut, terlihat bahwa rata-rata *jitter* paling besar pada algoritma penjadwalan WRR adalah pada saat menggunakan modulasi 64 QAM $\frac{3}{4}$, yaitu mencapai 8,269179 ms.

b. Pada skenario kedua :

- TRS_RR :



Gambar 4.8. Grafik Perbandingan *Throughput* TRS_RR pada berbagai Modulasi

Hasil dari simulasi dengan algoritma penjadwalan TRS_RR yang dilakukan pada modulasi QPSK $\frac{3}{4}$, 16 QAM $\frac{3}{4}$ dan 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.8. Berdasarkan gambar grafik *jitter* tersebut diatas, diperoleh nilai rata-rata *jitter* sebagai berikut :

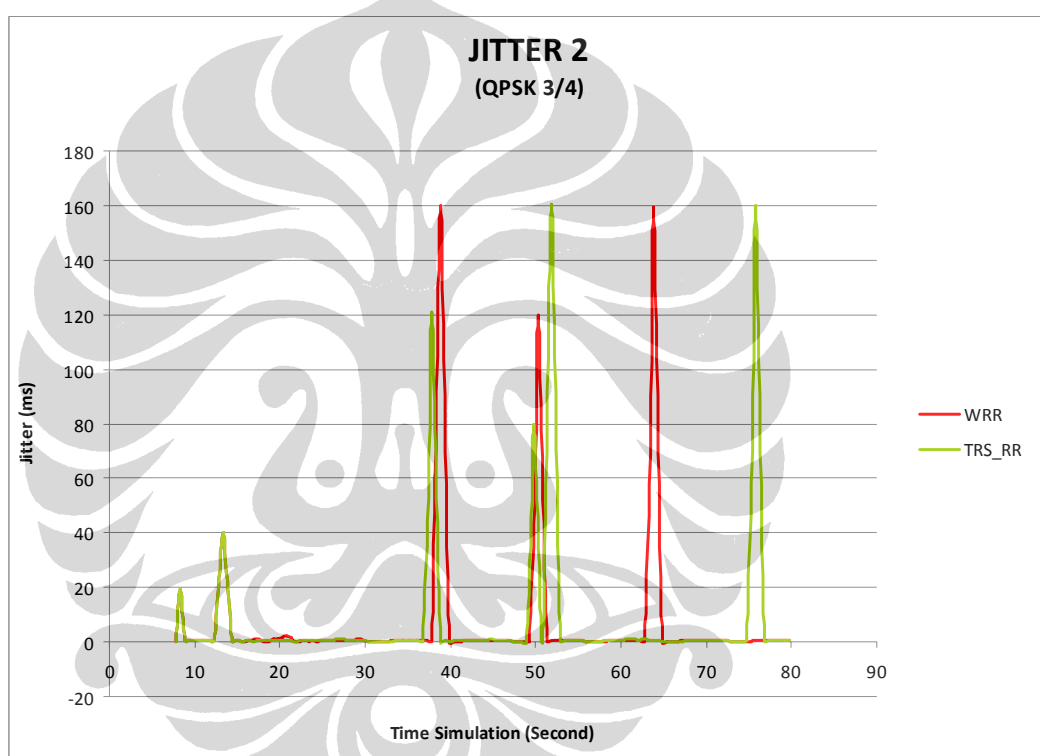
Average *Jitter* 2 :

TRS_RR (QPSK $\frac{3}{4}$)	:	8,321758621 ms.
TRS_RR (16QAM $\frac{3}{4}$)	:	8,29469 ms.
TRS_RR (64QAM $\frac{3}{4}$)	:	8,291731 ms.

Berdasarkan nilai tersebut, terlihat bahwa rata-rata *jitter* paling besar pada algoritma penjadwalan TRS_RR adalah pada saat menggunakan modulasi QPSK $\frac{3}{4}$, yaitu mencapai 8,321758621 ms.

c. Jika *jitter* WRR dan TRS_RR diperbandingkan pada modulasi yang sama :

- Modulasi QPSK $\frac{3}{4}$



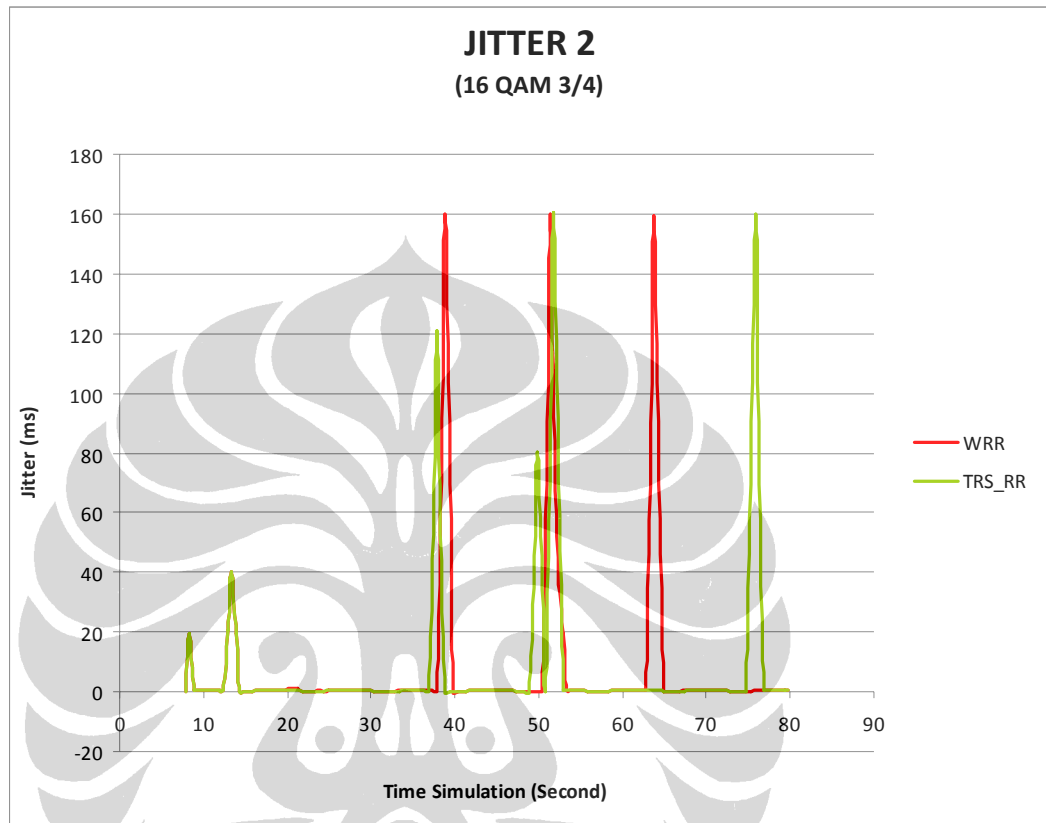
Gambar 4.9. Grafik Perbandingan *Jitter* WRR dan TRS_RR pada Modulasi QPSK $\frac{3}{4}$

Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi QPSK $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.9. Berdasarkan gambar grafik *jitter* tersebut diatas, diperoleh nilai rata-rata *jitter* sebagai berikut :

WRR (QPSK $\frac{3}{4}$)	:	7,215041 ms.
TRS_RR (QPSK $\frac{3}{4}$)	:	8,321758621 ms.

Berdasarkan nilai tersebut, terlihat bahwa pada modulasi QPSK $\frac{3}{4}$ TRS_RR memiliki rata-rata *jitter* lebih tinggi daripada WRR, yaitu sebesar 8,321758621 ms.

- Modulasi 16 QAM $\frac{3}{4}$



Gambar 4.10. Grafik Perbandingan *Jitter* WRR dan TRS_RR pada Modulasi 16 QAM $\frac{3}{4}$

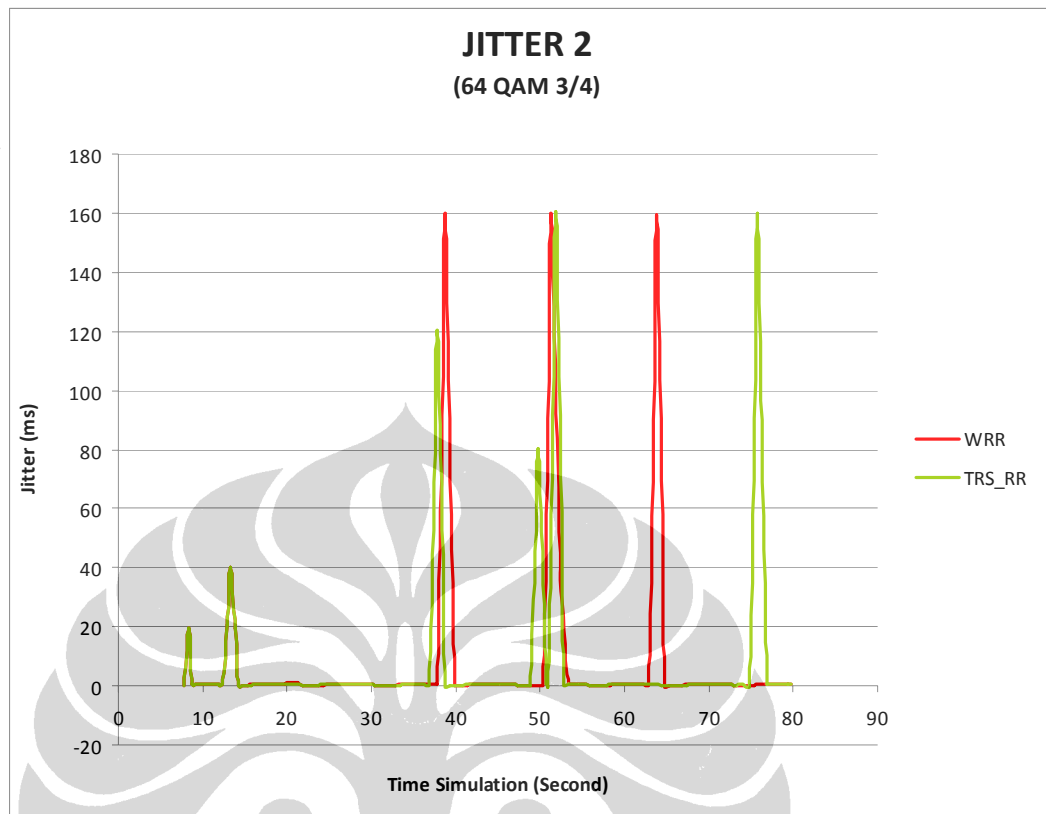
Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi 16 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.10. Berdasarkan gambar grafik *jitter* tersebut diatas, diperoleh nilai rata-rata *jitter* sebagai berikut :

WRR (16QAM $\frac{3}{4}$) : 8,269179 ms.

TRS_RR (16QAM $\frac{3}{4}$) : 8,29469 ms.

Berdasarkan nilai tersebut, terlihat bahwa rata-rata *jitter* pada modulasi 16 QAM $\frac{3}{4}$ TRS_RR memiliki rata-rata *jitter* lebih tinggi daripada WRR, yaitu mencapai 8,29469 ms.

- Modulasi 64 QAM 3/4



Gambar 4.11. Grafik Perbandingan *Jitter* WRR dan TRS_RR pada Modulasi 64 QAM 3/4

Hasil dari simulasi dengan algoritma penjadwalan WRR dan TRS_RR pada modulasi 64 QAM $\frac{3}{4}$ dibandingkan sebagaimana terlihat pada Gambar 4.11. Berdasarkan gambar grafik *jitter* tersebut diatas, diperoleh nilai rata-rata *jitter* sebagai berikut :

WRR (64QAM3/4) : 8,266724 ms.

TRS_RR (64QAM3/4) : 8,291731 ms.

Berdasarkan nilai tersebut, terlihat bahwa pada modulasi 64 QAM $\frac{3}{4}$ TRS_RR memiliki rata-rata *jitter* lebih tinggi daripada WRR, yaitu mencapai 8,291731 ms.

Berdasarkan pengamatan terhadap nilai rata-rata *jitter*, terlihat bahwa :

- Semakin tinggi modulasi yang digunakan, ternyata memberikan pengaruh yang berbeda terhadap algoritma penjadwalan WRR dan TRS-RR. Pada algoritma WRR, semakin tinggi modulasi yang digunakan, rata-rata *jitter* juga

cenderung semakin besar. Sebaliknya pada TRS_RR, semakin tinggi modulasi yang digunakan, rata-rata *jitter*nya cenderung semakin menurun;

- algoritma penjadwalan WRR memiliki nilai *jitter* tertinggi pada modulasi 16 QAM $\frac{3}{4}$ yaitu sebesar 8,269179 ms;
- algoritma penjadwalan TRS_RR memiliki nilai *jitter* tertinggi pada modulasi QPSK $\frac{3}{4}$ yaitu sebesar 8,321758621 ms;
- Meskipun pada modulasi yang semakin tinggi, rata-rata *jitter* TRS_RR semakin menurun, namun secara keseluruhan terlihat bahwa nilai rata-rata *jitter* TRS-RR selalu lebih tinggi daripada WRR pada setiap modulasi.

