BAB IV
HASIL EKSPERIMEN DAN PEMBAHASAN

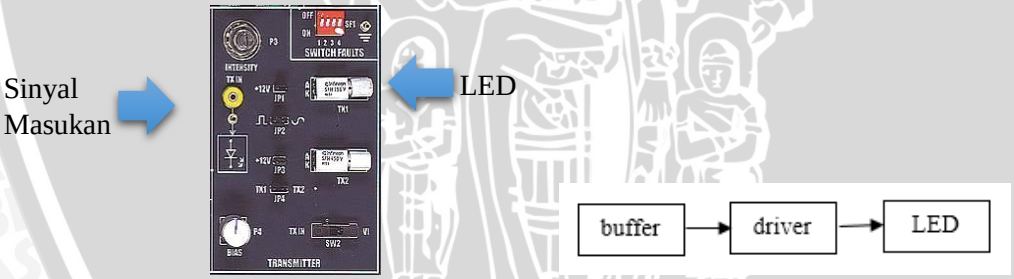
Bab IV menyajikan hasil dan pembahasan dari penelitian yang dilakukan. Data yang disajikan dari hasil penelitian meliputi pengaruh perubahan kecepatan yang divariasikan terhadap *delay, throughput*, dan *eye pattern*. Sistematika penyajian Bab IV terdiri atas konfigurasi perangkat-perangkat yang digunakan pada eksperimen, prosedur pengambilan data, dan analisis data yang telah didapat dari eksperimen.

4.1. Konfigurasi Perangkat Penelitian dan Prosedur

Perangkat penelitian terdiri dari spesifikasi alat-alat pengukuran yang dipergunakan, konfigurasi rancangan dari pengukuran, dan prosedur pengambilan data. Spesifikasi dan penjelasan perangkat eksperimen diuraikan sebagai berikut.

1) LED SFH756V

Perangkat FCL-03 memiliki satu pemancar LED tipe SFH756V. Cahaya keluaran LED berpusat pada gelombang merah tampak dengan panjang gelombang 660 nm. Gambar 4.1. menunjukkan perangkat *transmitter* pada FCL-03.



(a) (b)

Gambar 4.1. Perangkat *Transmitter* pada FCL-03
(Sumber: e-Manual Falcon *Advance Fiber Optic Communication Lab*, 2011)

Modul pemancar menerima sinyal masukan dalam bentuk elektrik lalu mengubahnya menjadi sinyal optik yang membawa informasi yang sama. Serat optik adalah media yang membawa sinyal optik tersebut menuju penerima.

Pemancar serat optik terdiri dari *buffer, driver,* dan sumber cahaya. Bagian *buffer* menyediakan sambungan elektrik dan isolasi di antara pemancar optik dan

sumber informasi. Bagian *driver* menyediakan catu daya optik ke sumber cahaya dengan menggandakan pola data yang dicatu ke pemancar. Kemudian sumber cahaya (LED) mengubah sinyal elektrik ke sinyal optik dengan pola yang sama.

2) *Plastic Optical Fiber* (POF)

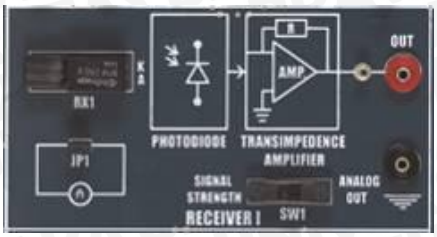
Serat optik yang digunakan adalah *Plastic Optical Fiber* (POF) yang terbuat dimana lapisan *core* terbuat dari *Polymethyl Methacrylate* (PMMA) sedangkan lapisan *cladding* dibuat dari *Perfluoropolimer*. Serat optik ini memiliki nilai *Numerical Aperture* (NA) sebesar 0,47. Diameter *core* sebesar 1 mm dan diameter *cladding* sebesar 2,20 mm. Nilai indeks biasnya adalah 1,492 pada *core* dan 1,417 pada *cladding*. Gambar 4.2. menunjukkan *plastic optical fiber*.

Gambar 4.2. *Plastic Optical Fiber*
(Sumber: e-Manual Falcon *Advance Fiber Optic Communication Lab*, 2011)

3) *Photodetector* SFH551/1V

Photodetector yang digunakan pada penelitian ini adalah tipe SFH551/1V, yaitu *phototransistor* dengan keluaran TTL *logic* yang berfungsi untuk mengubah kembali sinyal optik yang diterima dari serat optik untuk membentuk sinyal elektrik seperti pada pemancar. Ketika cahaya masuk ke detektor maka keluarannya TTL *high* sedangkan ketika tidak ada cahaya maka TTL *low*.

Photodetector terdiri dari *photodiode*, *transimpedance amplifier*, dan *level shifter*. *Transimpedance amplifier* digunakan untuk mengubah arus ke tegangan. Tegangan tersebut lalu dikuatkan dengan bantuan rangkaian penguat. Tegangan inilah yang merupakan duplikasi sinyal elektrik yang dipancarkan pada *transmitter*. Gambar 4.3. menunjukkan perangkat *receiver* pada FCL-03.

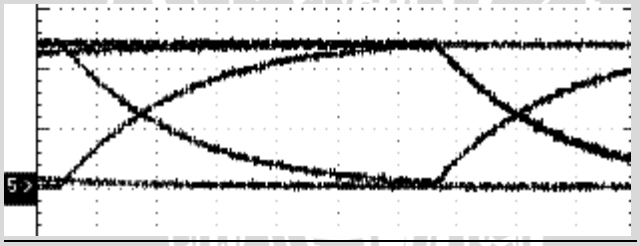


Gambar 4.3. Perangkat *Receiver* pada FCL-03
(Sumber: e-Manual Falcon *Advance Fiber Optic Communication Lab*, 2011)

4) Eye pattern generator

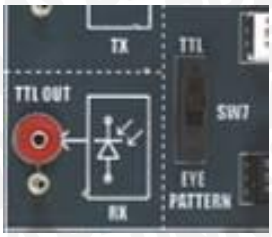
Eye pattern merupakan teknik yang mudah dan tepat untuk mengukur kemampuan sistem transmisi digital dalam menyalurkan data. Metode ini banyak digunakan untuk mengevaluasi kineja sistem berbasis kabel dan juga dapat digunakan untuk saluran data serat optik. Pengukuran *eye pattern* dilakukan dalam domain waktu dan memperbolehkan efek distorsi gelombang agar dapat diamati pada tampilan osiloskop.

Eye pattern dapat diamati menggunakan perangkat dasar yang tersedia. Keluaran dari CN2 masuk ke masukan vertikal osiloskop dan data digunakan untuk memicu *horizontal sweep*. Kedua masukan ini akan menghasilkan tampilan seperti pada Gambar 4.4.



Gambar 4.4. Tampilan *Eye Pattern* pada Osiloskop
(Sumber: e-Manual Falcon *Advance Fiber Optic Communication Lab*, 2011)

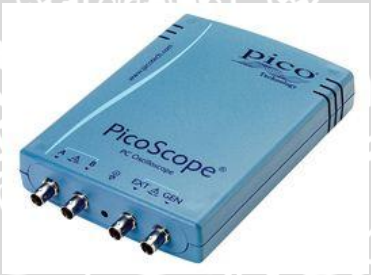
Gambar 4.4. disebut dengan *eye pattern*, karena bentuknya yang menyerupai mata manusia. Pola tersebut terbentuk dari kombinasi 4-*bit-long* NRZ yang ditumpangkan secara bersamaan. Gambar 4.5. menunjukkan perangkat *Eye Pattern* pada FCL-04.



Gambar 4.5. Perangkat *Eye Pattern* pada FCL-04
(Sumber: e-Manual Falcon *Advance Fiber Optic Communication Lab*, 2011)

5) Osciloscope

Pada percobaan ini digunakan PC *Oscilloscope* 60MHz jenis PicoScope 2207. Osiloskop ini memiliki dua kanal masukan dan mendukung *external trigger*. Tampilan osiloskop terhubung dengan PC sehingga memudahkan proses penyimpanan dan pencetakan tampilan sinyal. Untuk penggunaan pada modul *Fiber Optic Communication Lab*, kanal masukan pada osiloskop harus *ac coupled*, kecuali jika disebutkan. Gambar 4.6 menunjukkan perangkat PicoScope 2207.



Gambar 4.6. PicoScope 2207
(Sumber: www.picotech.com, 2014)

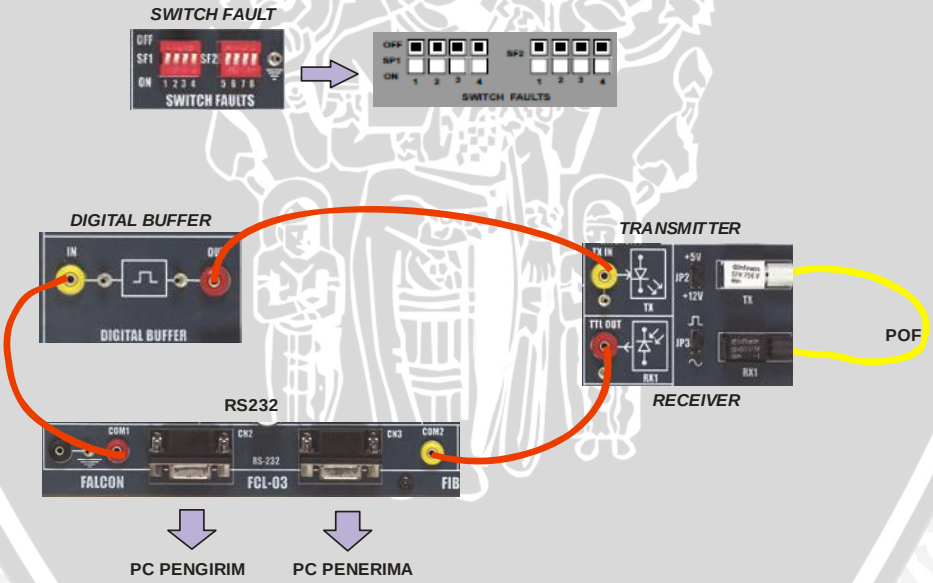
Prosedur Pengukuran

Prosedur pengambilan data melalui eksperimen diawali dengan mempersiapkan alat dan komponen pendukung pengukuran. Langkah pertama sebelum memulai pengukuran adalah membersihkan POF dengan menggunakan alkohol 95% dan kapas. Hal ini bertujuan menghilangkan kotoran dari ujung POF yang nantinya akan terhubung pada *phototransmitter* dan *photodetector*. Gambar 4.7 menunjukkan cara membersihkan ujung POF dengan menggunakan alkohol.

Gambar 4.7. Cara Membersihkan POF

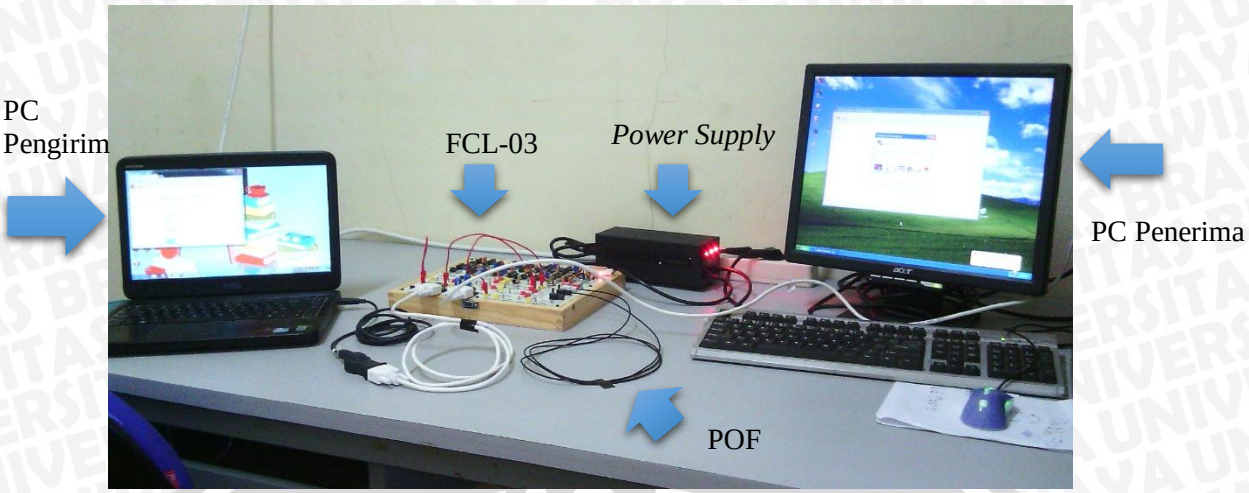
1) Merangkai Blok Diagram Pengukuran *Throughput* dan *Delay*

Proses pengukuran *throughput* dan *delay* hanya menggunakan satu perangkat inti, yaitu FCL-03. Pada modul FCL-03, antar muka yang ada adalah RS232. Antar muka RS232 menggunakan perangkat lunak Hyperterminal sebagai *user interface* yang berfungsi untuk komunikasi data antara dua PC. Gambar 4.8 menunjukkan diagram pengawatan FCL-03 yang digunakan untuk pengukuran *throughput* dan *delay*.



Gambar 4.8 Diagram Pengawatan pada FCL-03

Setelah semua perangkat yang dibutuhkan siap, maka langkah selanjutnya adalah merangkai perangkat tersebut sesuai dengan blok diagram perencanaan. Gambar 4.9 menunjukkan konfigurasi pengukuran *throughput* dan *delay* yang dilakukan pada penelitian ini.



Gambar 4.9. Konfigurasi Pengukuran *Throughput* dan *Delay*

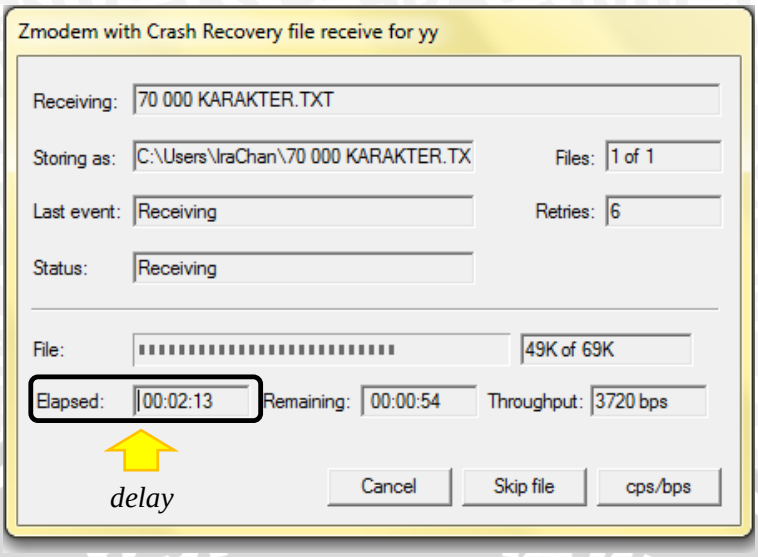
2) Melakukan Pengukuran *Delay* dan *Throughput*

Pengukuran *delay* dan *throughput* dilakukan dengan perangkat lunak Hyperterminal. Parameter pertama yang diukur adalah *delay* atau waktu transmisi, yaitu waktu yang dibutuhkan untuk mengirimkan suatu data keseluruhan dari PC pengirim ke PC penerima. Waktu *delay* yang terukur digunakan sebagai dasar untuk perhitungan *throughput*.

Pengambilan data dengan menggunakan Hyperterminal diawali dengan membuat koneksi antara dua PC yang saling berkomunikasi dengan pengaturan sesuai Tabel 4.1. Kemudian PC A mengirimkan *file* dengan format *.txt* dan PC B memilih menu untuk menerima file dari PC A. Protokol yang digunakan pada pengiriman file adalah Z Modem. Gambar 4.10 menunjukkan proses pengiriman *file* pada Hyperterminal. Nilai *delay* ditunjukkan oleh *time elapsed* saat pengiriman *file* telah selesai dilakukan.

Tabel 4.1 Pengaturan Parameter pada Hyperterminal

Bits per second	2400 4800 9600 19200 38400
Data Bit	8
Parity Bit	None
Start Bit	1
Flow Control	Xon/Xoff

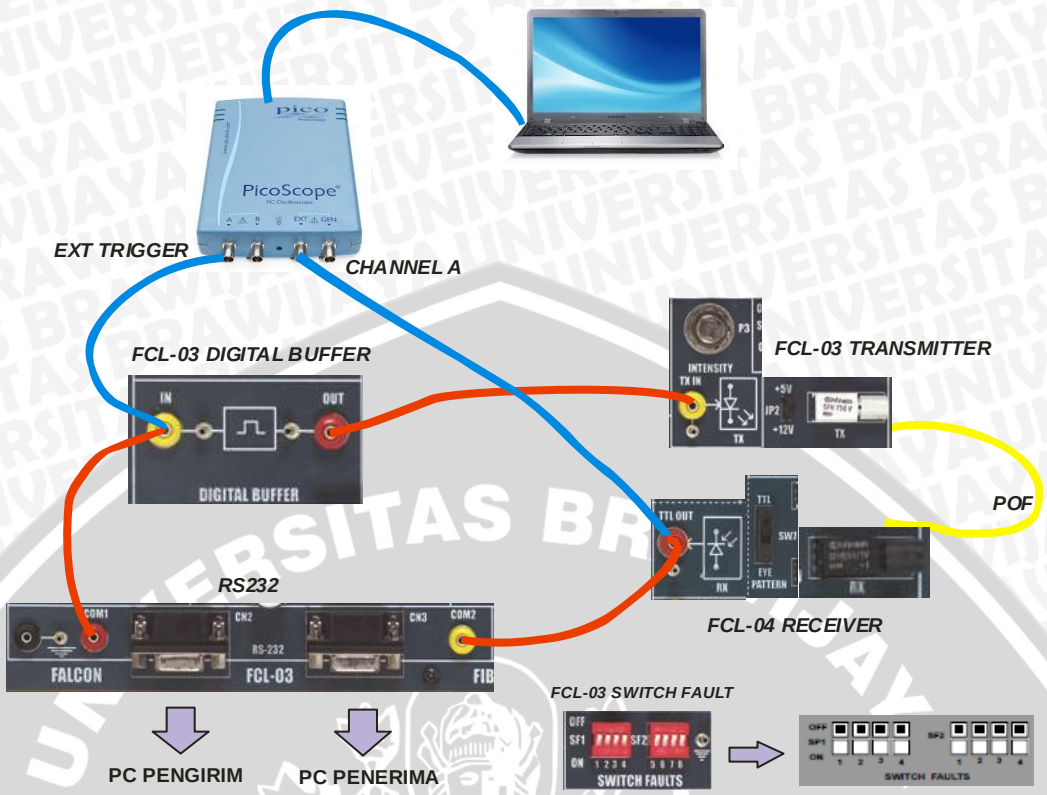


Gambar 4.10. Pengukuran Delay Menggunakan Hyperterminal

Nilai *throughput* didapatkan dengan membagi jumlah bit keseluruhan yang dikirimkan dengan *time elapsed* yang terukur pada Hyperterminal. Nilai *throughput* yang terukur merupakan nilai *throughput* rata-rata selama pengiriman data berlangsung dalam satuan bps.

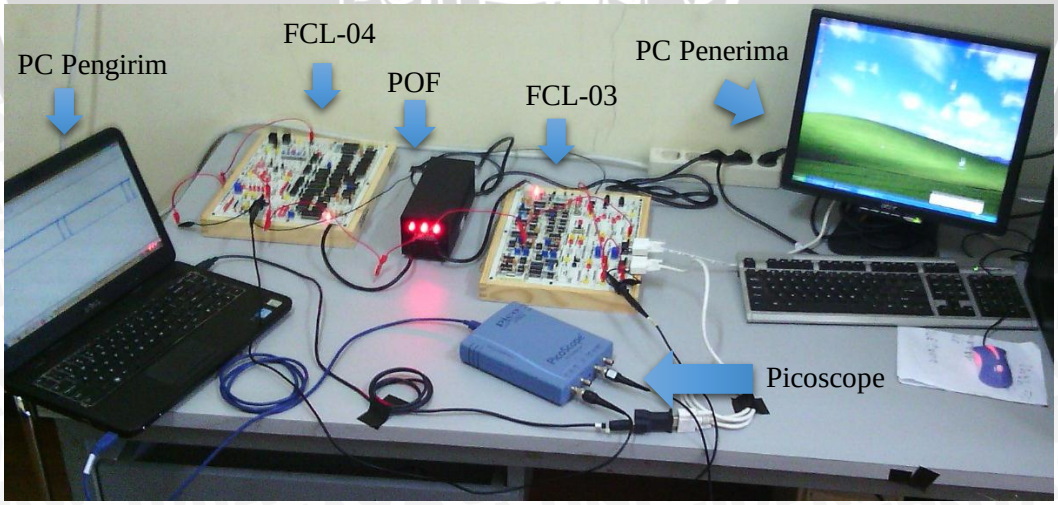
3) Merangkai Blok Diagram Pengukuran *Eye Pattern*

Proses pengukuran *eye pattern* menggunakan dua perangkat inti, yaitu FCL-03 dan FCL-04. Pada perangkat FCL-03, bagian yang digunakan adalah antar muka RS232, *digital buffer*, dan pemancar optik. Sedangkan bagian pada perangkat FCL-04 yang digunakan adalah penerima optik dan pembangkit *eye pattern*. Gambar 4.11 menunjukkan diagram pengawatan dalam pengukuran *eye pattern.*



Gambar 4.11 Diagram Pengawatan Pengukuran *Eye Pattern*

Setelah semua perangkat yang dibutuhkan siap, maka langkah selanjutnya adalah merangkat perangkat tersebut sesuai dengan blok diagram perencanaan. Gambar 4.12. menunjukkan konfigurasi pengukuran *eye pattern* yang dilakukan pada penelitian ini.



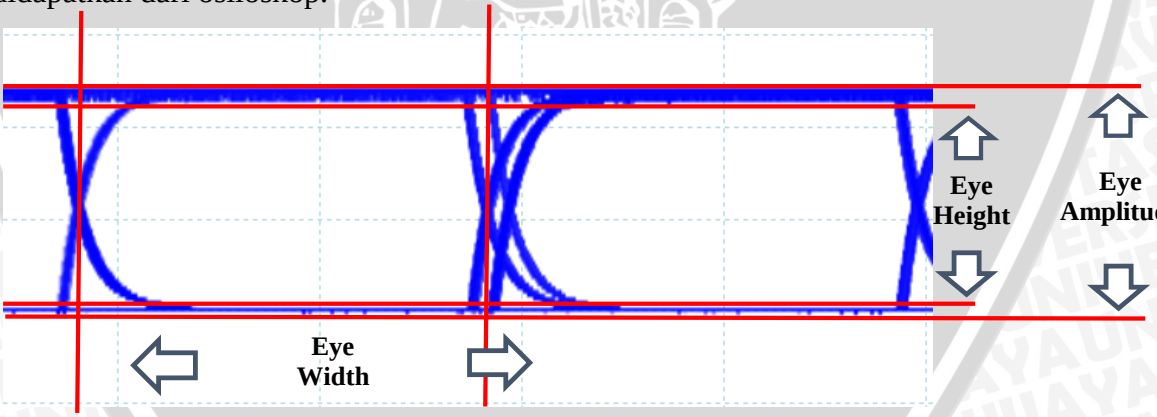
Gambar 4.12 Konfigurasi Pengukuran *Eye Pattern*

4) Melakukan Pengukuran *Eye Pattern*

Eye pattern merupakan salah satu cara untuk menganalisa kinerja suatu sistem transmisi digital. Tampilan dari osiloskop dari sinyal digital suatu sistem diolah proses *sampling* beberapa kali untuk mendapatkan karakteristik sinyal. Diperlukan software picoscope 6.0 untuk menampilkan keluaran sinyal dari picoscope sehingga pengukuran variabel *eye pattern* dapat dilakukan.

Pengukuran performansi suatu sistem dengan menggunakan *eye pattern* dapat dibagi menjadi dua bagian, yaitu *vertical eye opening* dan *horizontal eye opening*. Pada *vertical eye opening* terdapat dua parameter yaitu, *eye height* dan *eye amplitude*. *Eye height* merupakan besarnya *vertical eye opening* pada *eye pattern*, sedangkan *eye amplitude* menunjukkan daya sinyal informasi yang ditransmisikan dengan tidak memperhitungkan *noise*. Besarnya perbedaan antara *eye height* dan *eye amplitude* menunjukkan besarnya *noise* pada suatu sinyal. Semakin besar perbedaan *eye height* dan *eye amplitude*, maka semakin besar *noise* yang ada pada suatu sinyal.

Pada *horizontal eye opening*, terdapat *eye width* yang merupakan bukaan terlebar pada *eye opening*. *Horizontal eye opening* menunjukkan ketepatan *timing* pada proses transmisi data. Gambar 4.13 menunjukkan tampilan *eye pattern* yang didapatkan dari osiloskop.



Gambar 4.13 Tampilan *Eye Pattern* pada Osiloskop

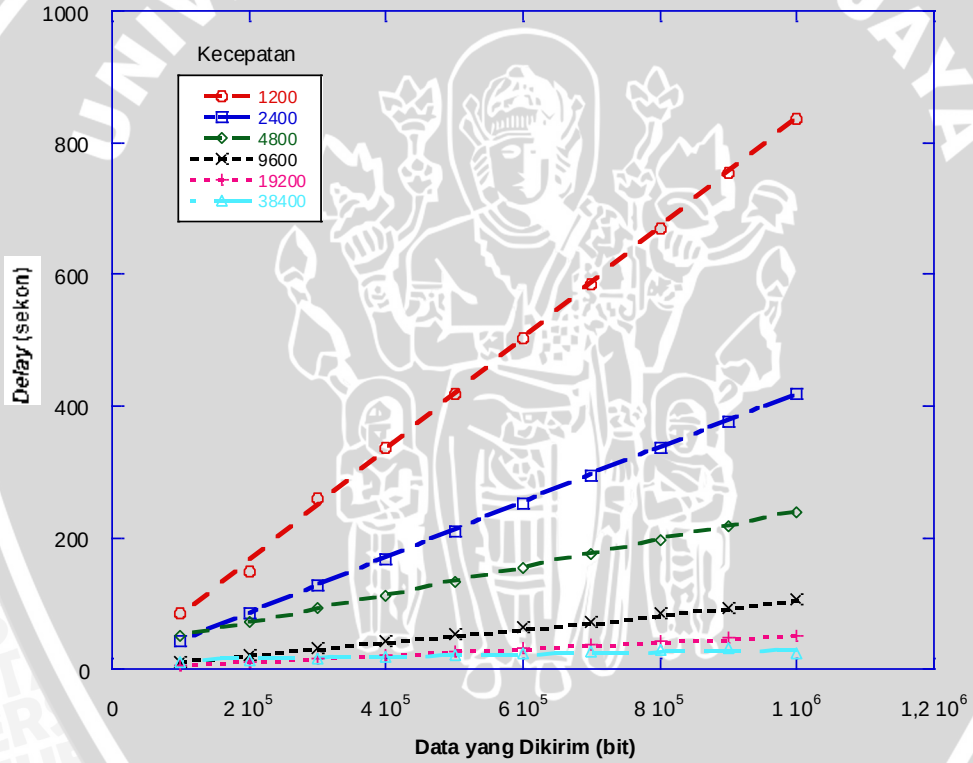
4.2. Data dan Pembahasan

Pada sub bab ini akan dibahas tentang perhitungan data yang didapat dari pengukuran dengan menggunakan kecepatan yang bervariasi untuk melihat

pengaruhnya terhadap performansi POF sebagai media komunikasi data. Analisis yang dilakukan meliputi *throughput*, *delay*, dan *eye pattern*. Metode yang dilakukan adalah metode matematis menggunakan data primer yang didapat dari hasil pengukuran di Laboratorium Telekomunikasi Universitas Brawijaya.

4.2.1. Analisis Pengaruh Perubahan Kecepatan terhadap *Delay*

Nilai *delay* diukur dengan menggunakan perangkat lunak Hyperterminal. Nilai *delay* menunjukkan durasi waktu yang dibutuhkan untuk mengirimkan suatu data secara keseluruhan dalam satuan sekon. Gambar 4.14 menunjukkan grafik *delay* untuk berbagai variasi *bit rate* dan jumlah data dikirim. Data *delay* dalam bentuk tabel ditampilkan pada Lampiran 1.



Gambar 4.14 Grafik *Delay* terhadap Jumlah Data Dikirim pada Berbagai Kecepatan

Nilai *delay* untuk seluruh kecepatan semakin naik seiring dengan naiknya jumlah data yang dikirim. Untuk kecepatan 1.200 bps, *delay* pada jumlah data dikirim sebesar 1 x 10^4 bit adalah 84 sekon. Sedangkan untuk jumlah data 1x10^6 bit, *delay* yang terukur adalah 836 sekon. Hal ini terjadi karena dengan semakin banyaknya data yang

dikirim, maka semakin banyak waktu yang dibutuhkan untuk mengirimkan data tersebut. Jika dibandingkan dengan kecepatan yang digunakan, nilai *delay* berbanding terbalik dengan kecepatan yang digunakan. Untuk jumlah data 1 x 10^6 bit, *delay* untuk kecepatan 1.200 bps adaah 836 sekon sedangkan *delay* untuk kecepatan 38.400 bps adalah 24 sekon. Nilai *delay* akan semakin kecil dengan semakin besarnya kecepatan yang digunakan. Dengan kecepatan yang semakin besar, maka akan lebih banyak data yang mampu dikirimkan tiap detik. Sehingga akan semakin sedikit waktu yang dibutuhkan untuk mengirim data tersebut secara keseluruhan. *Delay* paling besar terjadi pada kecepatan 1.200 bps, *delay* naik secara linier dari 84 sekon hingga mencapai nilai tertinggi 836 sekon.

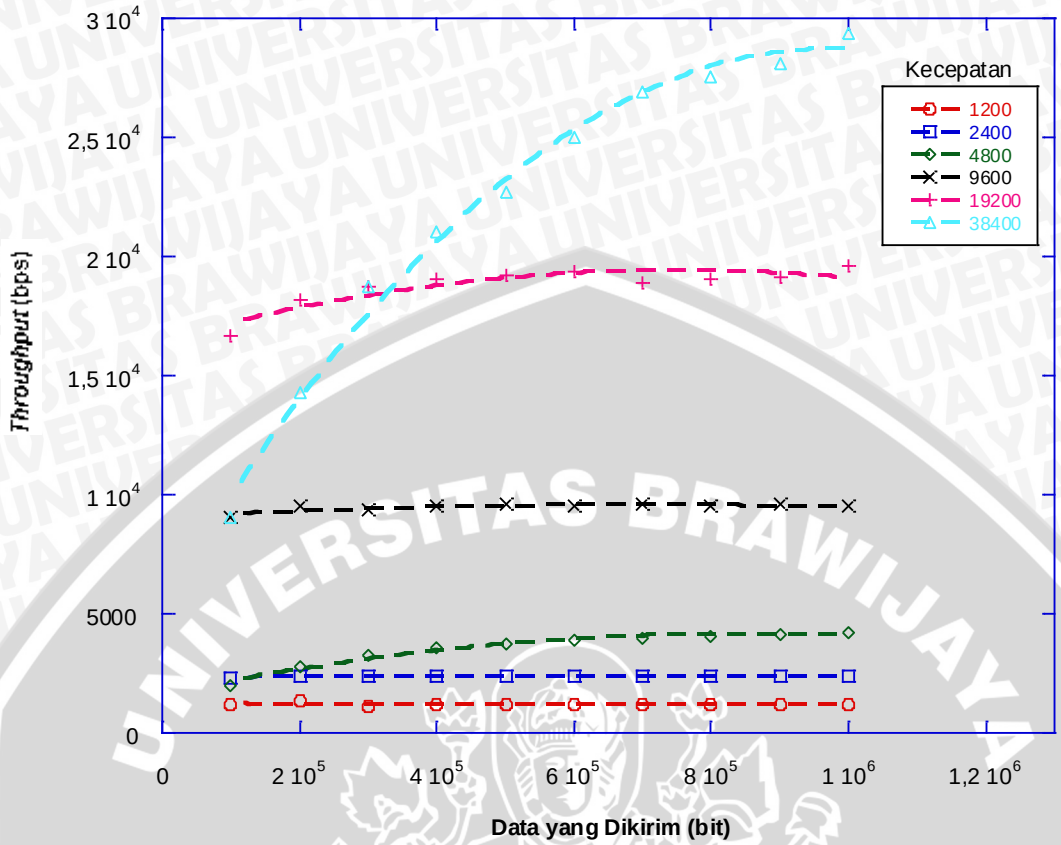
4.2.2. Analisis Pengaruh Perubahan Kecepatan terhadap *Throughput*

Proses pengukuran *throughput* dilakukan dengan menggunakan data *delay* yang sebelumnya telah didapatkan dari perangkat lunak Hyperterminal. *Throughput* yang terukur adalah *throughput* rata-rata yang dihitung selama pengiriman data berlangsung. Nilai *throughput* dapat dihitung dengan menggunakan Persamaan (2.4).

Besarnya *throughput* untuk data 1 x 10^4 bit pada kecepatan 1200 bps adalah:

$$Throughput\ (bps) = \frac{1\ x\ 10^4}{84} = 1190\ bps$$

Grafik pengukuran *throughput* untuk berbagai kecepatan dan jumlah data ditunjukkan pada Gambar 4.15. Data *throughput* dalam bentuk tabel ditampilkan pada Lampiran 2.



Gambar 4.15 Grafik *Throughput* Hasil Pengukuran

Nilai *throughput* berbanding lurus dengan jumlah data yang dikirimkan. Pada kecepatan 38.400 bps, *throughput* pada jumlah data dikirim sebesar $1x10^4$ bit adalah 9.091 bps. Sedangkan untuk jumlah data $1x10^6$ bit, *throughput* yang terukur adalah 29.412 bps. Semakin banyak data yang dikirim, maka sistem akan mengirimkan data dengan jumlah bit maksimal yang diijinkan oleh sistem sesuai dengan kecepatan yang digunakan. Namun pada jumlah data $1x10^4$ bit, *throughput* bernilai relatif kecil jika dibandingkan dengan kecepatan yang digunakan. Hal ini terjadi karena data yang dikirimkan ukurannya jauh lebih kecil dibanding dengan kecepatan yang digunakan. Sehingga saat pengiriman data telah selesai saat nilai *throughput* belum naik mencapai nilai maksimal.

Dari Gambar 4.15, dapat dilihat bahwa grafik *throughput* semakin naik seiring dengan semakin besarnya kecepatan yang digunakan. Namun untuk kecepatan 1.200, 2.400, 4.800 dan 9.600 bps, nilai *throughput* bersifat stabil untuk semua variasi jumlah data yang dikirim. Nilai *throughput* untuk empat kecepatan tersebut memiliki nilai

yang hampir sama dengan kecepatan yang digunakan. Hal ini menandakan kestabilan pengiriman data untuk ketiga kecepatan tersebut.

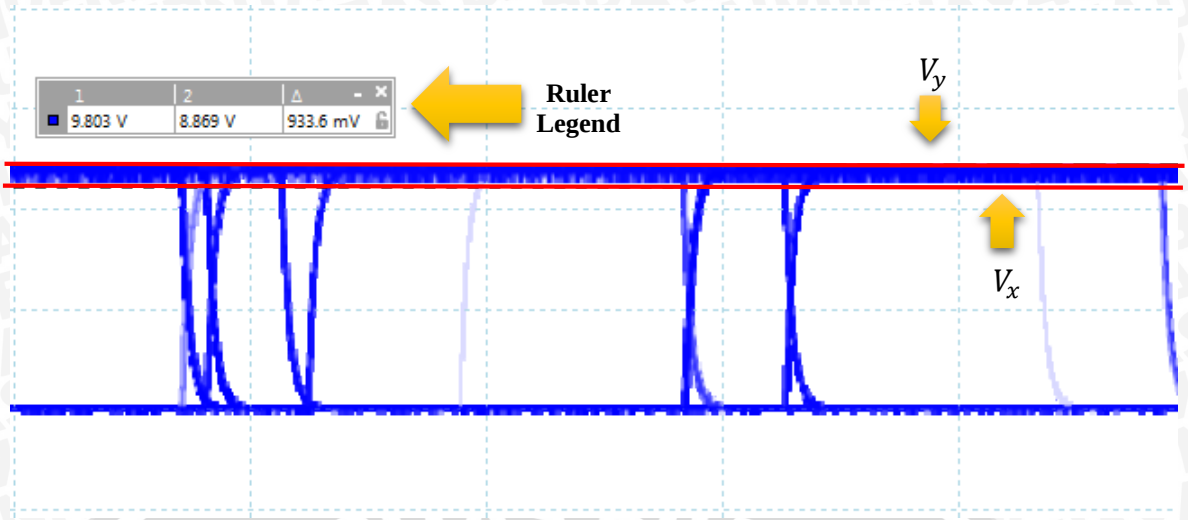
Sedangkan pada kecepatan 38.400 bps, *throughput* mengalami kenaikan secara signifikan. Grafik *throughput* tidak selalu bersifat linier disebabkan oleh perbedaan waktu proses *handshaking* pada komunikasi serial yang bersifat asinkron. Metode asinkron menggunakan *start bit* sebagai penanda awal sebuah data dikirim dan menggunakan *stop bit* sebagai tanda akhir suatu data. Apabila penerima tidak dapat mendeteksi *start bit* yang dikirimkan oleh pengirim, maka penerima akan mengirimkan sinyal berupa *bad data packet* dan pengirim akan mengirimkan ulang data yang hilang tersebut. Dengan adanya kondisi tersebut, maka diperlukan waktu tambahan untuk pengiriman kembali data yang hilang, sehingga nilai *throughput* akan menurun.

4.2.3. Analisis Pengaruh Perubahan Kecepatan terhadap *Eye Pattern*

Performansi serat optik yang selanjutnya dianalisis adalah *eye pattern*. *Eye pattern* adalah tampilan osiloskop yang digunakan untuk menganalisis sinyal digital. Pada *eye pattern* terhadap tiga parameter yang dapat dilihat yaitu *noise margin, timing jitter,* dan *bit rate*.

1. *Noise Margin*

Noise margin atau kekebalan terhadap *noise* ditunjukkan dengan panjang *eye opening* pada waktu *sampling*. *Noise margin* adalah ratio presentase dari puncak sinyal V_x yang dilihat dari besar *eye opening* terhadap tegangan maksimum sinyal V_y. Nilai *noise margin* yang diharapkan adalah yang tinggi karena semakin tinggi *noise margin* berarti sinyal tersebut semakin kebal terhadap *noise*. *Noise margin* dihitung dengan menggunakan persamaan (2.5) dan tampilan *eye pattern* untuk *noise margin* ditunjukkan pada Gambar 4.16. Data *noise margin* dalam bentuk tabel ditampilkan pada Lampiran 3.



Gambar 4.16 *Eye Pattern* pada Osiloskop

Besarnya *noise margin* untuk data 1 x 10^4 bit pada kecepatan 1200 bps adalah:

$$Noise\ Margin\ (\%) = \frac{8{,}983}{9.871} x\ 100\% = 91{,}004\%$$

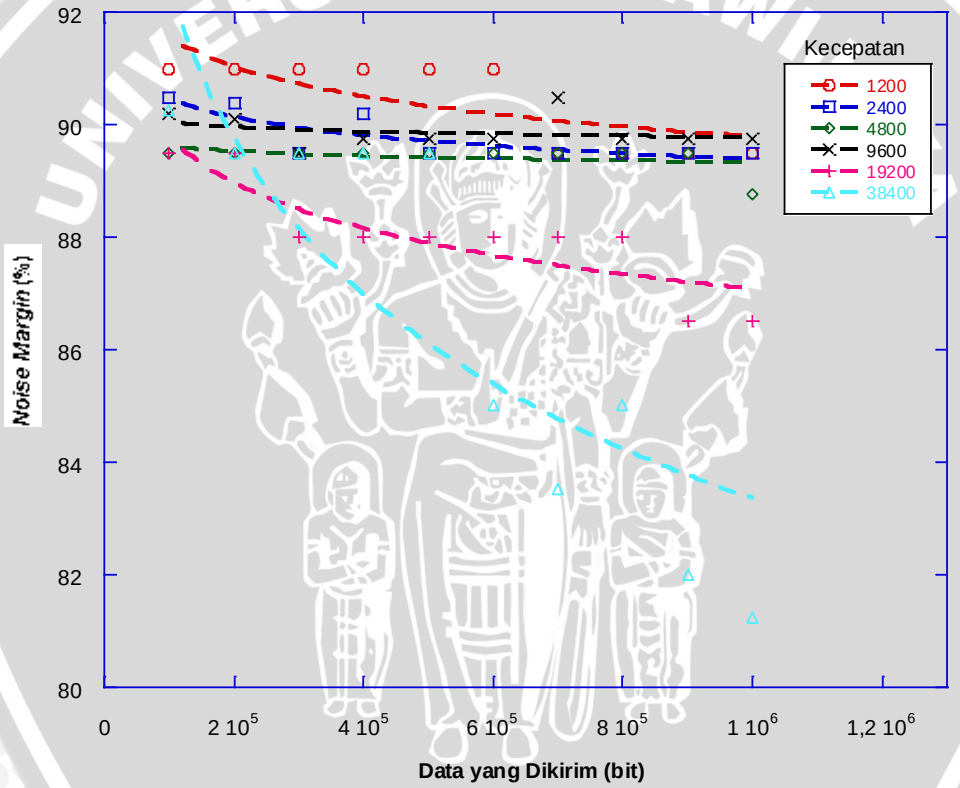
Dengan cara yang sama besar *noise margin* (dalam satuan %) untuk berbagai variasi kecepatan dan jumlah data ditunjukkan pada Gambar 4.17.

Nilai *noise margin* berbanding terbalik dengan besarnya jumlah data yang dikirim. Untuk kecepatan 38.400 bps, nilai *noise margin* untuk data 1x10^4 bit adalah 90,254% sedangkan *noise margin* untuk data 1x10^6 bit adalah 81,248%. Semakin banyak jumlah data yang dikirim maka nilai *noise margin* semakin menurun. Hal ini terjadi karena adanya jumlah data yang semakin besar, maka *throughput* juga akan semakin besar. *Throughput* yang besar menyebabkan periode bit akan semakin pendek sehingga *noise* akan dengan mudah mengubah data bit tersebut sehingga terjadi *error*. Sehingga dengan adanya jumlah data yang besar, gangguan *noise* akan semakin signifikan pengaruhnya terhadap data yang dikirimkan. Hal ini berbeda apabila kecepatan yang rendah yang digunakan. Untuk kecepatan 1.200 bps, nilai *noise margin* tidak berubah secara signifikan dengan semakin besarnya jumlah data yang dikirimkan. Nilai *noise margin* mengalami penurunan dari 91,004% menjadi 89,505%.

Jika dibandingkan dengan nilai variasi kecepatan, nilai *noise margin* juga berbanding terbalik dengan kecepatan yang digunakan. Untuk data 1x10^6 bit , pada kecepatan 1.200 bps nilai *noise margin* adalah 89,505% sedangkan pada kecepatan

38.400 bps nilai *noise margin* adalah 81,248%. Semakin besar kecepatan yang digunakan, nilai *noise margin* semakin turun. Turunnya nilai *noise margin* menandakan kekebalan sinyal terhadap *noise* semakin berkurang.

Penurunan nilai *noise margin* disebabkan oleh banyaknya jumlah data yang dikirimkan tiap detik, sehingga sinyal data yang dikirimkan akan semakin rapat dan semakin besar kemungkinan data yang akan rusak karena adanya pengaruh *noise*. *Noise* pada sinyal menyebabkan distorsi pada amplitudo sinyal yang menyebabkan variasi amplitudo sinyal. Variasi amplitudo sinyal ini mengakibatkan level tegangan bit '0' dan '1' semakin kecil sehingga mengakibatkan nilai *noise margin* yang semakin kecil.



Gambar 4.17 Grafik *Noise Margin* Hasil Pengukuran

2. *Timing Jitter*

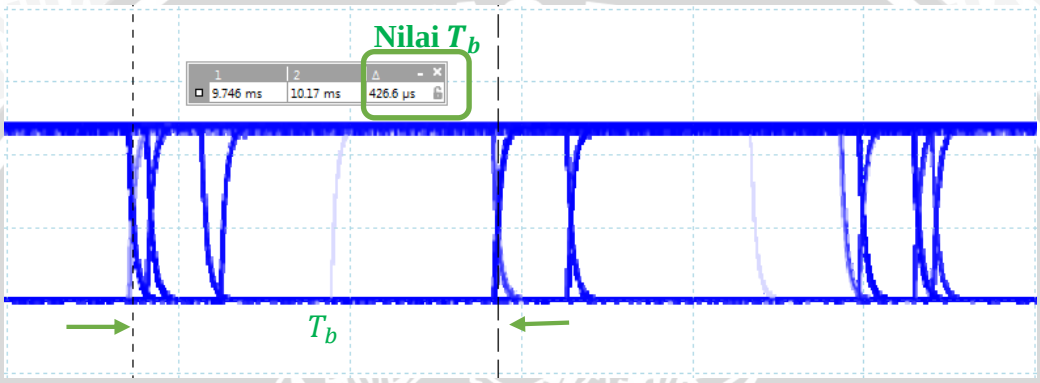
Timing jitter adalah penyimpangan waktu dari waktu ideal sebuah *event* data bit dan mungkin salah satu karakteristik yang paling penting dari sinyal data digital berkecepatan tinggi. Untuk menghitung *timing jitter*, penyimpangan waktu transisi dari naik dan turunnya tepi sebuah *eye pattern* pada titik persimpangan diukur.

Timing jitter dapat dihitung dari jumlah distorsi (ΔT) dan *bit interval* (T_b) dengan persamaan (2.7) dan tampilan *eye pattern* untuk *timing jitter* ditunjukkan pada Gambar 4.18 dan Gambar 4.19. Data *timing jitter* dalam bentuk tabel ditampilkan pada Lampiran 4.

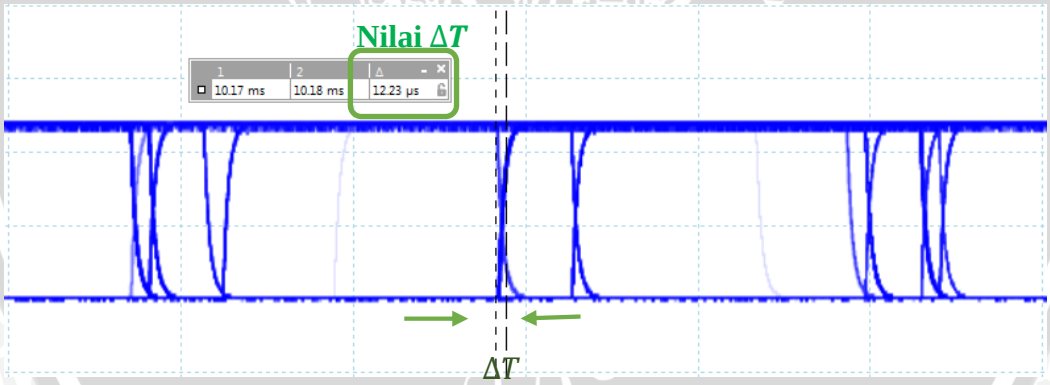
Besarnya *timing jitter* untuk data 1 x 10^4 bit pada kecepatan 1200 bps adalah:

$$Timing\ Jitter\ (\%) = \frac{\Delta T}{T_b} x100\% = \frac{12{,}23}{833{,}4} x\ 100\% = 1{,}467\%$$

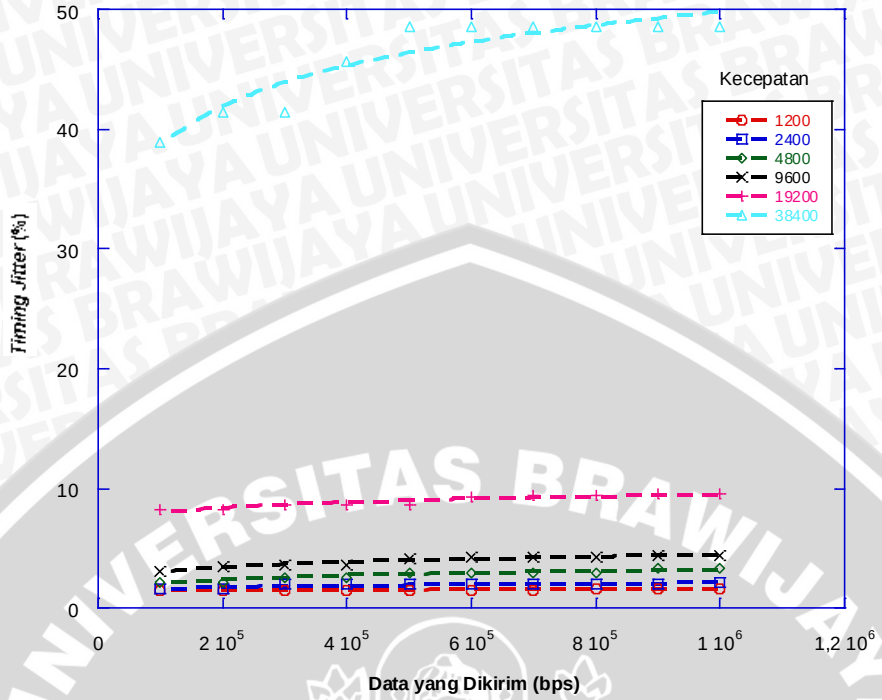
Dengan cara yang sama besar *timing jitter* (dalam satuan %) untuk berbagai variasi kecepatan dan jumlah data ditunjukkan pada Gambar 4.20.



Gambar 4.18 *Bit Interval* pada *Eye Pattern*



Gambar 4.19 Distorsi Waktu pada *Eye Pattern*



Gambar 4.20 Grafik *Timing Jitter* Hasil Pegukuran

Nilai *timing jitter* berbanding lurus jika dibandingkan dengan jumlah data yang dikirimkan. Pada kecepatan 1.200 bps, untuk data $1x10^4$ bit nilai *timing jitter* adalah 1,4675% sedangkan untuk data $1x10^6$ bit nilai timing jitter adalah 1,6540%. Dengan semakin besarnya jumlah data yang dikirimkan, maka waktu untuk tiap *event* dapat mengalami perbedaan yang semakin besar karena adanya tumbukan antar data yang semakin besar.

Nilai *timing jitter* berbanding lurus dengan besarnya kecepatan yang digunakan. Untuk data $1x10^6$ bit, pada kecepatan 1.200 bps nilai *timing jitter* adalah 1,6540% sedangkan pada kecepatan 38400 bps nilai *timing jitter* adalah 48,580%. Semakin besar kecepatan yang digunakan, nilai *timing jitter* semakin tinggi. Dengan kecepatan kecil, perbedaan waktu tiap *event* tidak terlalu besar karena nilai periode setiap bit cenderung tetap, sehingga nilai *timing jitter* rendah. Namun jika kecepatan yang digunakan semakin besar, maka *bit rate* menjadi tidak stabil dan periode untuk setiap bit berubah dan membuat setiap *event* mengalami perbedaan waktu yang cukup besar sehingga nilai *timing jitter* juga semakin besar. Nilai *timing jitter* yang besar ditandai dengan semakin sempitnya *horizontal eye*

opening pada *eye pattern*. Namun jika dilihat berdasarkan banyaknya data yang dikirimkan, nilai *timing jitter* cenderung stabil untuk berbagai jumlah data yang dikirimkan. Sehingga dapat diketahui bahwa nilai *timing jitter* lebih dipengaruhi oleh nilai kecepatan yang digunakan dan tidak terpengaruh oleh besarnya data yang dikirim.

3. *Bit Rate*

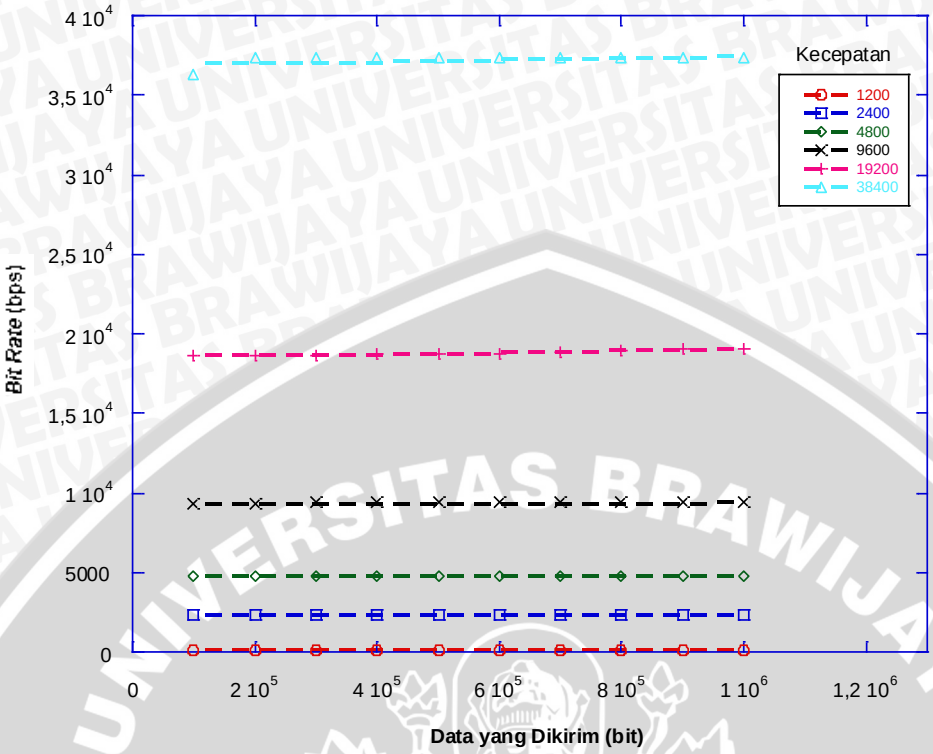
Bit period adalah ukuran dari pembukaan horizontal *eye pattern* pada titik-titik persimpangan mata. *Bit rate* adalah kebalikan dari periode bit (1/periode bit). Dengan menggunakan *bit period*, *bit rate* dapat ditentukan dengan persamaan (2.6). Tampilan *eye pattern* untuk *bit period* ditunjukkan oleh Gambar 4.18. Data *bit rate* dalam bentuk tabel ditampilkan pada Lampiran 5.

Besarnya *bit rate* untuk data 1 x 10^4 bit pada kecepatan 1200 bps adalah:

$$Bit\ Rate(bps) = \frac{1}{833{,}4\ x\ 10^{-6}} = 1200\ bps$$

Dengan cara yang sama besar *bit rate* (dalam satuan bps) untuk berbagai variasi kecepatan dan jumlah data ditunjukkan pada Gambar 4.21.

Nilai *bit rate* cenderung stabil untuk semua variasi jumlah data yang dikirim. Berapa pun jumlah data yang dikirim, nilai *bit rate* akan tetap sama. Pada kecepatan 1.200 bps, nilai bit rate untuk data 1 x 10^4 bit adalah 1.200 bps dan untuk data 1x10^6 bit adalah 1.200 bps. Sedangkan jika dibandingkan dengan kecepatan, nilai *bit rate* berbanding lurus dengan kecepatan yang digunakan. Semakin besar kecepatan yang digunakan, semakin besar jumlah *bit rate*. *Bit rate* tertinggi adalah 37.369 bps untuk kecepatan 38.400 bps dan nilai *bit rate* terendah adalah 1.200 bps untuk kecepatan 1.200 bps. Nilai *bit rate* selalu lebih kecil atau sama dengan nilai kecepatan yang digunakan. Hal ini menunjukkan bahwa kecepatan merupakan besaran yang menunjukkan nilai maksimal bit yang mampu dilewatkan pada tiap sekon.

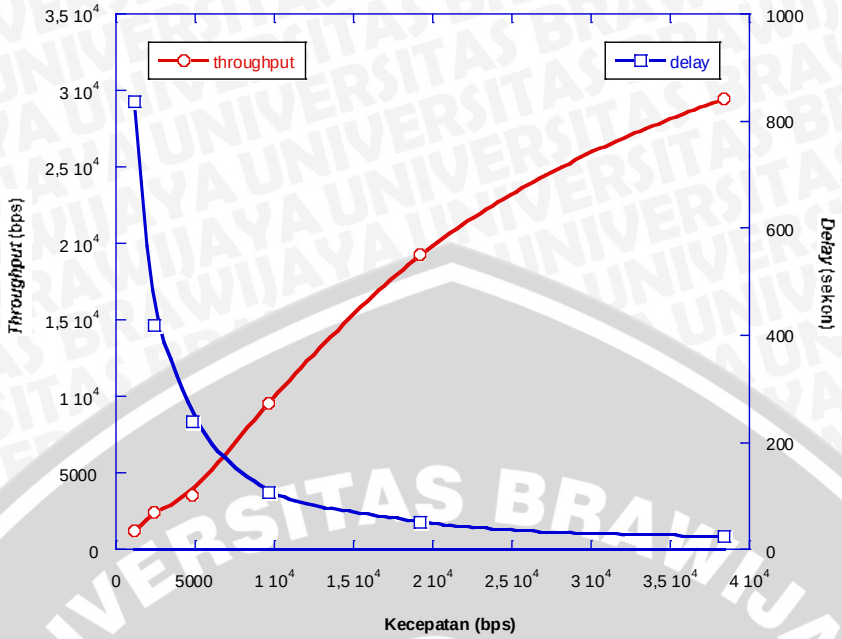


Gambar 4.21 Grafik *Bit Rate* Hasil Pegukuran

4.2.4. Hubungan *Throughput* dan *Delay*

Hubungan antara variasi kecepatan, *throughput*, dan *delay* untuk pengiriman data sebesar $1x10^6$ bit ditunjukkan dalam bentuk grafik seperti pada Gambar 4.22.

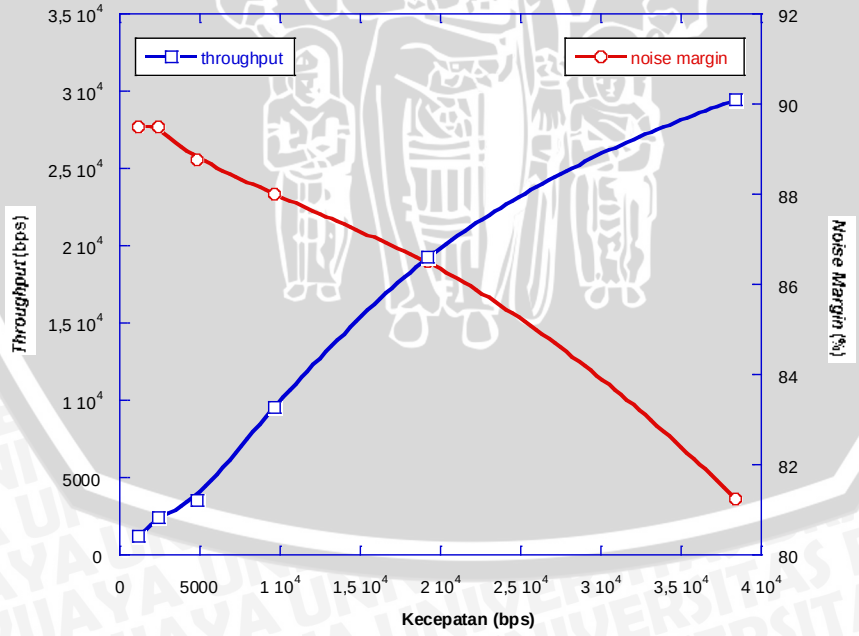
Berdasarkan grafik hubungan pada Gambar 4.22, terlihat bahwa perubahan nilai kecepatan memberikan konsekuensi perubahan terhadap nilai *throughput* dan *delay*. Pada kecepatan 1.200 bps, nilai *throughput* adalah 1.196 bps dan nilai *delay* adalah 836 sekon. Sedangkan untuk kecepatan 38.400 bps, nilai *throughput* adalah 29.412 bps dan nilai *delay* adalah 24 sekon. Dengan semakin besarnya nilai kecepatan yang digunakan maka nilai *throughput* semakin besar dan nilai *delay* semakin rendah. Nilai *throughput* yang semakin besar menandakan semakin banyaknya data yang dikirimkan tiap sekon, maka semakin sedikit waktu yang diperlukan untuk pengiriman seluruh data. Hal ini menunjukkan keterkaitan antara *throughput* dan *delay* yang saling berbanding terbalik.



Gambar 4.22 Hubungan Kecepatan, *Throughput*, dan *Delay*

4.2.5. Hubungan *Throughput* dan *Noise Margin*

Hubungan antara variasi kecepatan, *throughput*, dan *noise margin* untuk pengiriman data sebesar $1x10^6$ bit ditunjukkan dalam bentuk grafik seperti pada Gambar 4.23.

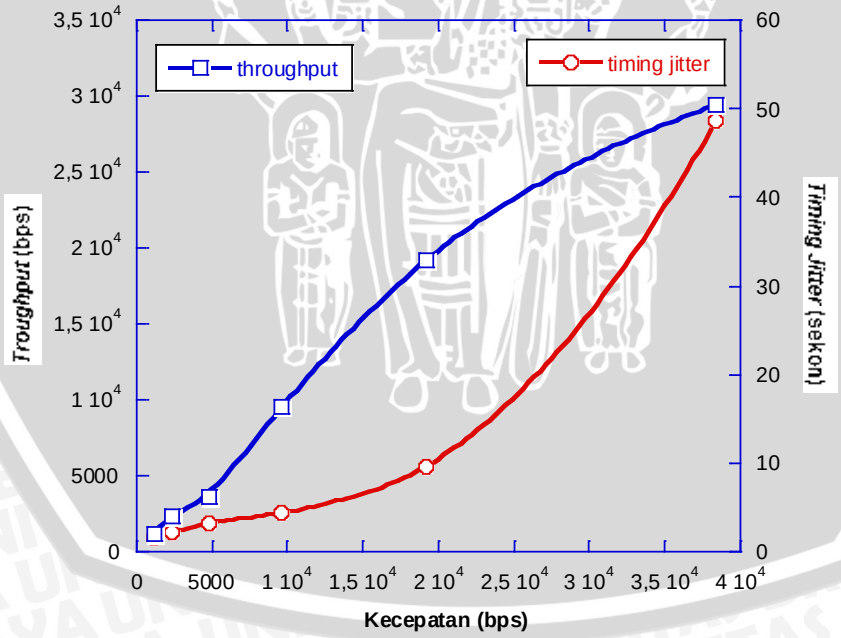


Gambar 4.23 Hubungan Kecepatan, *Throughput*, dan *Noise Margin*

Berdasarkan grafik hubungan pada Gambar 4.23, terlihat bahwa perubahan nilai kecepatan memberikan konsekuensi perubahan terhadap nilai *throughput* dan *noise margin*. Pada kecepatan 1.200 bps, nilai *throughput* adalah 1.196 bps dan nilai *noise margin* adalah 89,505%. Sedangkan untuk kecepatan 38.400 bps, nilai *throughput* adalah 29.412 bps dan nilai *noise margin* adalah 81,248%. Dengan semakin besarnya nilai kecepatan yang digunakan maka nilai *throughput* semakin besar dan nilai *noise margin* semakin turun. Hal ini meunjukkan keterkaitan *throughput* yang semakin tinggi menyebabkan kekebalan sinyal terhadap *noise* menurun. *Throughput* yang tinggi mengakibatkan semakin banyaknya data yang dikirimkan pada tiap detik. Jumlah data yang semakin banyak menyebabkan data akan semakin berdekatan dalam pengirimannya dan apabila ada *noise* maka semakin banyak amplitudo dari data yang terganggu. *Throughput* dan *noise margin* memiliki hubungan berbanding terbalik.

4.2.6. Hubungan *Throughput* dan *Timing Jitter*

Hubungan antara variasi kecepatan, *throughput*, dan *timing jitter* untuk pengiriman data sebesar $1x10^6$ bit ditunjukkan dalam bentuk grafik seperti pada Gambar 4.24.

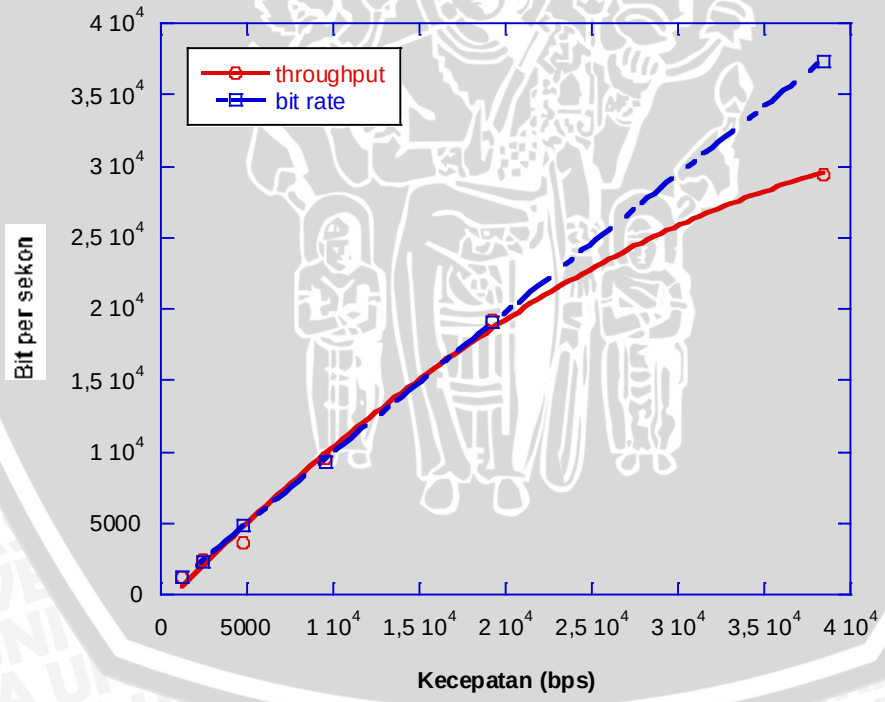


Gambar 4.24 Hubungan Kecepatan, *Throughput*, dan *Timing Jitter*

Berdasarkan grafik hubungan pada Gambar 4.24, terlihat bahwa perubahan nilai kecepatan memberikan konsekuensi perubahan terhadap nilai *throughput* dan *timing jitter*. Pada kecepatan 1.200 bps, nilai *throughput* adalah 1.196 bps dan nilai *timing jitter* adalah 1,654%. Sedangkan untuk kecepatan 38.400 bps, nilai *throughput* adalah 29.412 bps dan nilai *noise margin* adalah 48,580%. Dengan semakin besarnya nilai kecepatan yang digunakan, maka nilai *throughput* semakin besar dan mengakibatkan nilai *timing jiiter* semakin besar. Hal ini terjadi karena dengan semakin banyaknya data yang dilewatkan setiap detik, kemungkinan tumbukan antar data akan semakin besar. Tumbukan ini menyebabkan adanya perbedaan waktu tiba untuk setiap data sehingga muncul *timing jitter*. Gambar 4.24 menunjukkan hubungan antara *throughput* dan *timing jitter* yang berbanding lurus.

4.2.7. Hubungan *Throughput* dan *Bit Rate*

Hubungan antara variasi kecepatan, *throughput*, dan *bit rate* untuk pengiriman data sebesar 1 x 10^6 bit ditunjukkan dalam bentuk grafik seperti pada Gambar 4.25.



Gambar 4.25 Hubungan Kecepatan, *Throughput*, dan *Bit Rate*

Berdasarkan grafik hubungan pada Gambar 4.25, terlihat bahwa perubahan nilai kecepatan memberikan konsekuensi perubahan terhadap nilai *throughput* dan *bit*

rate. Dapat dilihat bahwa *throughput* berbanding lurus dengan *bit rate*. Nilai *throughput* dan *bit rate* memiliki nilai yang hampir sama untuk kecepatan 1.200, 2.400, 4.800, 9.600, dan 19.200 bps, dan mengalami perbedaan untuk kecepatan yang lebih besar dari 19.200 bps. Untuk kecepatan 1.200 bps nilai *throughput* adalah 1.196 bps sedangkan nilai bit rate adalah 1.200 bps. Sedangkan untuk kecepatan 9.600 bps, nilai *throughput* adalah 9.524 bps sedangkan nilai bit rate adalah 9.426 bps. Pada kecepatan 38.400, terjadi perbedaan yang cukup signifikan antara nilai *throughput* dan *bit rate*. Nilai *throughput* untuk kecepatan 38.400 adalah 29.412 bps sedangkan nilai *bit rate* adalah 37.369 bps. Pada dasarnya *throughput* dan *bit rate* merupakan besaran yang mengukur banyaknya data yang dilewatkan tiap detik. Namun apabila diteliti lebih jauh, *throughput* merupakan besaran yang mengukur rata-rata banyaknya bit per detik yang dilewatkan untuk suatu periode tertentu, yaitu selama pengiriman data berlangsung. *Throughput* dapat berbeda nilai dari satu waktu dengan waktu yang lainnya. Sedangkan *bit rate* diukur dari *eye pattern* yang merupakan proses sampling dari seluruh data yang dikirimkan, sehingga *bit rate* hanya didapatkan dari suatu waktu tertentu sehingga nilai bit rate dapat mengalami perbedaan dari satu waktu dengan waktu lainnya.