LAMPIRAN A

A.1 Ketika M *user* aktif, sinyal transmisi berhubungan dengan k^{th} bit data dari m^{th} user, sinyal transmisi tiap *subcarrier* pada user ke m^{th} :

$$s_{m,i}(t) = a_m[k]c_m[i]\cos(2\pi(f_c + i\frac{F}{Tb})t + \phi_i) \times P_{T_b}(t - kT_b) \qquad \text{(A.1)}$$

dimana $f_i = f_c + iF/Tb$, f_i merupakan frekuensi *subcarrier* dan ϕ_i adalah phasa dari *subcarrier*, maka sinyal trasmisi untuk setiap user :

$$s_m(t) = \sum_{i=0}^{N-1} c_m[i]a_m[k]\cos(2\pi f_i t + \phi_i) \times P_{T_b}(t - kT_b) \qquad \text{(A.2)}$$

Sinyal trasmisi keseluruhan yang dikirim adalah :

$$s(t) = \sum_{m=1}^{M} \sum_{i=0}^{N-1} c_m[i]a_m[k]\cos(2\pi f_i(t - \tau_m) + \phi_i) \times P_{T_b}(t - kTb - \tau_m) \qquad \text{(A.3)}$$

A.2 Setiap sinyal *subcarrier* pada m^{th} user yaitu:

$$r_{m,i}(t) = \sum_{l=0}^{L-1} \rho_l c_i^m(t - \tau_{m,l})a_k^m(t - \tau_{m,l})\cos(2\pi f_i(t - \tau_l - \tau_m) + (\phi_{m,i} - \varphi_l))$$

$$r_{m,i}(t) = \sum_{l=0}^{L-1} \rho_l c_i^m(t - \tau_{m,l})a_k^m(t - \tau_{m,l})\cos(2\pi f_i t + \theta_{m,i,l}) \qquad \text{(A.4)}$$

dimana $\theta_{m,i,l} = \phi_{m,i} - \varphi_l - 2\pi f(\tau_l + \tau_m)$, adalah *independent and identically distributed* (iid) *uniform random variabel* pada interval [0,2π],dan sinyal user ke m^{th} di penerima:

$$r_m(t) = \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} \rho_l c_i^m(t - \tau_m - \tau_l)a_k^m(t - \tau_m - \tau_l)\cos(2\pi f_i t + \theta_{m,i,l}) \qquad \text{(A.5)}$$

A.3 Diasumsikan pada user ke m=0, dan $\tau_{m=0}$=0, sama dengan sinyal yang diinginkan dengan koreksi fasa sempurna, $\hat{c}_i^0(t) = c_i^0(t - \tau_l)$ pada lintasan l^{th}, $\theta_{0,i,l} = \hat{\theta}_{0,i}$ menunjukkan estimasi fasa penerima pada i^{th} *subcarrier*:

$$v = \int_{kTb}^{(k+1)Tb} r(t)\frac{2}{Tb}d_{m,i}\,\hat{c}_i^m(t)\cos(2\pi f_i t + \hat{\theta}_{m,i})\,dt \qquad \text{(A.6)}$$

$$v = \sum_{m=1}^{M} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} \rho_{i,l}^{m} c_i^m (t-\tau_m - \tau_l) a_k^m (t-\tau_m - \tau_l) \cos(2\pi f_i t + \theta_{m,i,l}) \times \int_{kTb}^{(k+1)Tb} \frac{2}{Tb} d_{m,i} \hat{c}_i^m (t) \cos(2\pi f_i t + \hat{\theta}_{m,i}) \, dt + \eta$$

$$\eta = \sum_{i=0}^{N-1} \int_{kTb}^{(k+1)Tb} n(t) \frac{2}{Tb} d_{m,i} \cos(2\pi f_i t + \hat{\theta}_{m,i}) \, dt$$

Untuk sinyal yang diinginkan tanpa noise :

$$v_0 = \sum_{i=0}^{N-1} \sum_{i=0}^{L-1} \rho_{i,l}^{0} \; \hat{c}_i^0 (t)^2 \; d_{0,i} \; \hat{a}_k^0 (t) \; \frac{2}{Tb} \int_{kTb}^{(k+1)Tb} \cos^2 (2\pi f_i t + \hat{\theta}_{0,i}) \, dt = \sum_{i=0}^{N-1} \sum_{i=0}^{L-1} \rho_{i,l}^{0} \, d_{0,i} \hat{a}_k^0 (t) \qquad \text{(A.7)}$$

A.4 Diketahui untuk $0 \le t \le Tb$, $a(t) = a_k^1[0]$ dan $[c_v^l(t)]^2 = 1$, sehingga dapat disederhanakan menjadi:

$$D_{v,lp} = \int_0^{T_b} (\rho_{lp} c_v^1 (t) a_k^1 (t) \cos(2\pi f_v t + \theta_{1,v,lp}) \bullet \beta_{v,lp} c_v^1 (t) 2 / T_b \cos(2\pi f_v t + \theta_{v,lp}^1) dt \qquad \text{(A.8)}$$

$$D_{v,lp} = \rho_{lp} \beta_{v,lp} a_k^1 (t) [c_v^1 (t)]^2 \int_0^{T_s} 2 / T_b \cos^2 (2\pi f_v t + \theta_{v,lp}^1) dt = \rho_{lp} \beta_{v,lp} \; a_k^1 [0]$$

A.5 *Interference* oleh *L multipath* dapat disederhanakan menjadi :

$$I_1^S = \int_0^{Tb} \sum_{l=0, l \ne lp}^{L-1} \rho_l c_v^1 (t - \tau_{1,l}) a_k^1 (t - \tau_{1,l}) \cos(2\pi f_v t + \theta_{1,v,l}) \bullet \beta_{v,lp} c_v^1 (t) 2 / T_b \cos(2\pi f_v t + \theta_{v,lp}^1) dt \qquad \text{(A.9)}$$

$$I_1^S = \sum_{l=0, l \ne lp}^{L-1} I_{1,l}^S$$

$$I_{1,l}^S = \rho_l \beta_{v,lp} a_k^1 (t - \tau_{1,l}) c_v^1 (t - \tau_{1,l}) c_v^1 (t) \bullet \int_0^{Tb} 1 / T_b (\cos(4\pi f_v t + \theta_{1,v,l} + \theta_{v,lp}^1) + \cos(\theta_{v,lp}^1 - \theta_{1,v,l})) dt \qquad \text{(A.10)}$$

Untuk f_v >>1/T_S, maka term *cos*(4π…) akan berharga sama dengan nol. Sedangkan $a_k^1(t-\tau_{1,l}) = a_k^1[-1]$ pada $0 \le t < \tau_{1,l}$,dan $a_k^1(t-\tau_{1,l}) = a_k^1[0]$ pada $\tau_{1,l} \le t \le T_b$ sehingga

$$I_{1,l}^S = \rho_l \beta_{v,lp} \cos(\theta_{v,lp}^1 - \theta_{1,v,l})\{a_k^1[-1]R_1(\tau_{1,l}) + a_k^1[0]R_1(\tau_{1,l})\}/T_b \qquad \text{(A.11)}$$

dimana $R_1(\tau_{1,l})$ merupakan fungsi *partial autocorelation* yang didefinisikan :

$$R_1(\tau_{1,l}) = \int_0^{\tau_{1,l}} c_v^1(t-\tau_{1,l})c_v^1(t)dt \text{ dan}$$
$$\hat{R}_1(\tau_{1,l}) = \int_{\tau_{1,l}}^{Tb} c_v^1(t-\tau_{1,l})c_v^1(t)dt \qquad \text{(A.12)}$$

A.6 *Interference* oleh *subcarrier* yang lain dapat disederhanakan menjadi :

$$I_2^S = \int_0^{Tb} \sum_{i=0,i\neq v}^{N-1} \sum_{l=0}^{L-1} \rho_l c_i^1(t-\tau_{1,l}) a_k^1(t-\tau_{1,l}) \cos(2\pi f_i t + \theta_{1,i,l}) \qquad \text{(A.13)}$$
$$\bullet\, \beta_{v,l} c_v^1(t) 2/T_b \cos(2\pi f_v t + \theta_{v,l}^1)dt$$
$$I_2^S = \sum_{i=0,i\neq v}^{N-1} \sum_{l=0}^{L-1} I_{2,i}^S$$

$$I_{2,l}^S = \rho_l \beta_{i,l} a_k^1(t-\tau_{1,l}) c_i^1(t-\tau_{1,l}) c_v^1(t) \bullet 1/T_b$$
$$\int_0^{Tb} (\cos(2\pi(f_i + f_v)t + \theta_{1,i,l} + \theta_{v,lp}^1) + \cos(2\pi(f_i - f_v)t + \theta_{1,i,l} - \theta_{v,lp}^1))dt \qquad \text{(A.14)}$$
$$I_{2,l}^S = \rho_l \beta_{i,l}\{a_k^1[-1]R_1(\tau_{1,l},\gamma_{i,l}) + a_k^1[0]\hat{R}_1(\tau_{1,l},\gamma_{i,l})\}/T_b$$

dimana $R_1(\tau_{1,l}, \gamma_{i,l})$ merupakan fungsi *partial crosscorelation* yang didefinisikan :

$$R_1(\tau_{1,l},\gamma_{i,l}) = \int_0^{\tau_{1,l}} c_i^1(t-\tau_{1,l})c_v^1(t)\cos(2\pi(f_i - f_v)t + \gamma_{i,l})dt$$
$$\hat{R}_1(\tau_{1,l},\gamma_{i,l}) = \int_{\tau_{1,l}}^{Tb} c_i^1(t-\tau_{1,l})c_v^1(t)\cos(2\pi(f_i - f_v)t + \gamma_{i,l})dt \qquad \text{(A.15)}$$

Fungsi $\gamma_{i,l} = (\theta_{1,i,l} - \theta_{1,v,l})$ dan perbedaan *subcarrier* didefinisikan; $f_i - f_v = \lambda(i-v)/T_b$.

A.7 *Interference* oleh komponen *multipath user* yang lain pada *subcarrier* yang sama dapat disederhanakan menjadi :

$$I_{1,l}^{M} = \rho_{v,l}^{m}\beta_{v,lp}a_{k}^{m}(t-\tau_{m,l})c_{v}^{m}(t-\tau_{m,l})c_{v}^{1}(t) \tag{A.16}$$

$$\bullet \int_{0}^{Tb} 1/T_b(\cos(4\pi f_v t + \theta_{m,v,l} + \theta_{v,lp}^{1}) + \cos(\theta_{v,lp}^{1} - \theta_{m,v,l}))dt$$

$$I_{1,l}^{M} = \rho_{v,l}^{m}\beta_{v,lp}\{a_{k}^{m}[-1]R_m(\tau_{m,l}) + a_{k}^{m}[0]\hat{R}_m(\tau_{m,l})\}/T_b$$

dimana $R_m(\tau_{m,l})$ merupakan fungsi *partial crosscorelation* yang didefinisikan :

$$R_m(\tau_{m,l}) = \int_{0}^{\tau_{1,l}} c_{i}^{m}(t-\tau_{m,l})c_{v}^{1}(t)\cos(\theta_{v,lp}^{1} - \theta_{m,v,l})dt$$

$$\hat{R}_m(\tau_{m,l}) = \int_{\tau_{1,l}}^{Tb} c_{i}^{m}(t-\tau_{m,l})c_{v}^{1}(t)\cos(\theta_{v,lp}^{1} - \theta_{m,v,l})dt \tag{A.17}$$

A.8 *Interference* oleh komponen *multipath user* yang lain pada *subacarrier* yang berbeda dapat disederhanakan menjadi :

$$I_{2,l}^{M} = \rho_{i,l}^{m}\beta_{v,lp}a_{k}^{m}(t-\tau_{m,l})c_{i}^{m}(t-\tau_{m,l})c_{v}^{1}(t) \bullet \int_{0}^{Tb} 1/T_b \tag{A.18}$$

$$(\cos(2\pi(f_i + f_v)t + \theta_{m,i,l} + \theta_{v,lp}^{1}) + \cos(2\pi(f_i - f_v)t + \theta_{v,lp}^{1} - \theta_{m,v,l}))dt$$

$$I_{2,l}^{M} = \rho_{i,l}^{m}\beta_{v,lp}\{a_{k}^{m}[-1]R_m(\tau_{m,l},\gamma_{m,i,l}) + a_{k}^{m}[0]\hat{R}_m(\tau_{m,l},\gamma_{m,i,l})\}/T_b$$

dimana $R_m(\tau_{m,l}\ ,\ \gamma_{m,i,,l}\ )$ merupakan fungsi *partial crosscorelation* yang didefinisikan :

$$R_1(\tau_{m,l},\gamma_{m,i,l}) = \int_{0}^{\tau_{1,l}} c_{i}^{m}(t-\tau_{m,l})c_{v}^{1}(t)\cos(2\pi(f_i - f_v)t + \gamma_{m,i,l})dt$$

$$\hat{R}_m(\tau_{m,l},\gamma_{m,i,l}) = \int_{\tau_{1,l}}^{Tb} c_{i}^{m}(t-\tau_{m,l})c_{v}^{1}(t)\cos(2\pi(f_i - f_v)t + \gamma_{m,i,l})dt \tag{A.19}$$

Dan fungsi $\gamma_{m,i,,l} = (\theta_{m,i,l}-\theta_{1,v,l})$.

A.9 Melihat persamaan [A.5] , [A.6], [A.7] dan [A.8] merupakan kasus khusus dari $I_{2,l}^{M}$ yaitu:

- $I_{1,l}^{S} = I_{2,l}^{M}$ dengan $m = 1$, $i = v$
- $I_{2,l}^{S} = I_{2,l}^{M}$ dengan $m = 1$
- $I_{1,l}^{M} = I_{2,l}^{M}$ dengan $i = v$

Maka Var[$I_{2,l}^{M}$] didefinisikan terlebih dahulu kemudian *term* yang lain diturunkan sebagai kasus khusus.

$$Var[I_{2,l}^{M}] = E[(\rho_{i,l}^{m})^2](\beta_{v,lp}/T_b)^2\{E[(a_k^m[-1])^2 E\tau_{m,l},\gamma_{m,i,l}[R_m(\tau_{m,l},\gamma_{m,i,l})] + E[(a_k^m[0])^2 E\tau_{m,l},\gamma_{m,i,l}[\hat{R}_m(\tau_{m,l},\gamma_{m,i,l})]\} \quad (3.30)$$

berdasarkan *decision statistik*, dimana $\tau = \tau_{m,l}$, $x = \lambda(i\text{-}v)/T_b$, $N_e = T_b/T_b'$ dan $\phi = \gamma_{m,l,i}$, serta $E[(\rho_{i,l}^{m})^2] = \Omega_{i,l}^{m}$ merupakan MIP dari kanal, didapatkan [9] :

$$Var[I_{2,l}^{M}] = \Omega_{i,l}^{m}(\beta_{v,lp})^2 \frac{N_e}{2\pi^2(i-v)^2\lambda^2}\{1-\sin c[\frac{2(i-v)\lambda}{N_e}]\} \quad (A.20)$$

Kemudian diturunkan untuk $Var[I_{1,l}^{M}]$ dengan limit $i \rightarrow v$, dan menggunakan expansion series untuk sin(x) didapatkan:

$$Var[I_{1,l}^{M}] = \lim_{i-->v} Var[I_{2,l}^{M}] = \frac{\Omega_{v,l}^{m}(\beta_{v,lp})^2}{3N_e} \quad (A.21)$$

sama halnya untuk $Var[I_{2,l}^{S}] = \lim_{m->1} Var[I_{2,l}^{M}]$; didapatkan:

$$Var[I_{2,l}^{S}] = \Omega_{i,l}^{1}(\beta_{v,lp})^2 \frac{N_e}{2\pi^2(i-v)^2\lambda^2}\{1-\sin c[\frac{2(i-v)\lambda}{N_e}]\} \quad (A.22)$$

dan yang terakhir untuk $Var[I_{1,l}^{S}] = \lim_{S->V} Var[I_{1,2}^{S}]$; didapatkan:

$$Var[I_{1,l}^{S}] = \frac{\Omega_{i,l}^{1}(\beta_{v,lp})^2}{3N_e} \quad (A.23)$$

subsitusi persamaan (2.33) diperoleh :

$$\begin{aligned} Var[Z_{vlp}] = {} & \frac{N_o\beta_{v,lp}^2}{T_b} + \sum_{l=0,l\neq lp}^{L-1}\frac{\Omega_{v,l}^{m}(\beta_{v,lp})^2}{3N_e} \\ & + \sum_{i=0,i\neq v}^{N-1}\sum_{l=0}^{L-1}\Omega_{i,l}^{1}(\beta_{v,lp})^2\frac{N_e}{2\pi^2(i-v)^2\lambda^2}\{1-\sin c[\frac{2(i-v)\lambda}{N_e}]\} \\ & + \sum_{m=2}^{M}\sum_{l=0}^{L-1}\frac{\Omega_{v,l}^{m}(\beta_{v,lp})^2}{3N_e} \\ & + \sum_{m=2}^{M}\sum_{i=0,i\neq v}^{N-1}\sum_{l=0}^{L-1}\Omega_{i,l}^{m}(\beta_{v,lp})^2\frac{N_e}{2\pi^2(i-v)^2\lambda^2}\{1-\sin c[\frac{2(i-v)\lambda}{N_e}]\} \end{aligned} \quad (A.24)$$

berdasarkan exponential MIP,untuk semua lintasan dapat diset η =0 maka:

$$\sum_{l=0}^{L-1}\Omega_l = \Omega_0(1-\exp(-\eta L))/(1-\exp(-\eta)) \quad (A.25)$$

dan dinotasikan J_m sebagai ;

$$J_m = \sum_{i=0, i \neq v}^{N-1} \frac{N_e}{2\pi^2 (i-v)\lambda^2} \{1 - \sin c[\frac{2(i-v)\lambda}{N_e}]\} \tag{A.26}$$

maka persamaan di atas dapat disederhanakan :

$$\begin{aligned} Var[Z_{vlp}] = {} & \frac{N_o \beta_{v,lp}^2}{T_b} + M \frac{(\beta_{v,lp})^2}{3N_e} \sum_{l=0}^{L-1} \Omega_l - \frac{(\beta_{v,lp})^2}{3N_e} \Omega_{lp} \\ & + \sum_{i=0, i \neq v}^{N-1} (\beta_{v,lp})^2 \frac{N_e}{2\pi^2 (i-v)^2 \lambda^2} \{1 - \sin c[\frac{2(i-v)\lambda}{N_e}]\} \bullet M \sum_{l=0}^{L-1} \Omega_{lp} \end{aligned} \tag{A.27}$$

$$Var[Z_{vlp}] = \Omega_0 \beta_{v,lp}^2 \left[\left(\frac{2\Omega_0 T_b}{N_o} \right)^{-1} + M \frac{(1 - \exp(-\eta L))}{(1 - \exp(-\eta))} \left(\frac{1}{3N_e} + J_m \right) - \frac{\exp(-\eta lp)}{3N_e} \right]$$

LAMPIRAN B

Sifat statistik noise

Kalkulasi karakteristik statistik untuk EGC , adapun pendekatan matematis untuk persaman komponen *noise* sebagai berikut :

$$\eta = \sum_{i=0}^{N-1} \int_{kTb}^{(k+1)Tb} n(t) \frac{2}{Tb} d_{0,i} \cos(2\pi f_c t + 2\pi i \frac{F}{Tb} t + \theta_{0i})\, dt \qquad \text{(B.1)}$$

Persamaan di atas diasumsikan *large number of subcarrier* artinya nilai N besar, distribusi dari *noise*, η dapat didekati dengan distribusi Gaussian menurut *Central Limit Theorem* (CLT). Nilai tengah dari *noise*, η adalah sebagai berikut :

$$E\eta = \sum_{i=0}^{N-1} \int_{kTb}^{(k+1)Tb} En(t) \frac{2}{Tb} E d_{0,i}\, E\{\cos(2\pi f_c t + 2\pi i \frac{F}{Tb} t + \theta_{0i})\}\, dt = 0 \qquad \text{(B.2)}$$

variance noise, η dapat ditentukan sebagi berikut :

$$\sigma_\eta^2 = E\eta^2$$

$$= E\left\{\left[\sum_{i=0}^{N-1} \int_{kTb}^{(k+1)Tb} n(t_1) \frac{2}{Tb} d_{0,i} \cos(2\pi f_c t_1 + 2\pi i \frac{F}{Tb} t_1 + \theta_{0,i})\, dt_1\right] \times \left[\sum_{j=0}^{N-1} \int_{kTb}^{(k+1)Tb} n(t_2) \frac{2}{Tb} d_{0,j} \cos(2\pi f_c t_2 + 2\pi j \frac{F}{Tb} t_2 + \theta_{0,j})\, dt_2\right]\right\}$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \int_{kTb}^{(k+1)Tb} \int_{kTb}^{(k+1)Tb} \frac{4}{T_b^2} E\{n(t_1) n(t_2)\} E\{d_{0,i} d_{0,j}\} \times$$

$$E\{\cos(2\pi f_c t_1 + 2\pi i \frac{F}{T_b} t_1 + \hat{\theta}_{0,i}) \cos(2\pi f_c t_2 + 2\pi j \frac{F}{T_b} t_2 + \theta_{0,j}) dt_1\, dt_2$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \int_{kTb}^{(k+1)Tb} \int_{kTb}^{(k+1)Tb} \frac{4}{T_b^2} \frac{No}{2} \delta(t_1 - t_2) E\{d_{0,i} d_{0,j}\} \times$$

$$E\{\cos(2\pi f_c t_1 + 2\pi i \frac{F}{T_b} t_1 + \hat{\theta}_{0,i}) \cos(2\pi f_c t_2 + 2\pi j \frac{F}{T_b} t_2 + \theta_{0,j}) dt_1\, dt_2$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \int_{kTb}^{(k+1)Tb} \frac{4}{T_b^2} No\, E\{d_{0,i} d_{0,j}\} \times$$

$$E\{\cos(2\pi f_c t_1 + 2\pi i \frac{F}{T_b} t_1 + \theta_{0,i}) \cos(2\pi f_c t_1 + 2\pi j \frac{F}{T_b} t_1 + \theta_{0,j}) dt_1$$

$$= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{2}{T_b^2} NoE\{d_{0,i} d_{0,j}\} \times$$

$$E \int_{kTb}^{(k+1)Tb} \left[\frac{1}{2} \cos(2\pi f_c t_1 + 2\pi F \frac{i-j}{T_b} t_1 + \theta_{0,i} - \theta_{0,j}) + \frac{1}{2} \cos(4\pi f_c t_1 + 2\pi F \frac{i-j}{T_b} t_1 + \theta_{0,i} + \theta_{0,j} \right] dt_1$$

$$\cong \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{2}{T_b^2} NoE\{d_{0,i} d_{0,j}\} E \int_{kTb}^{(k+1)Tb} \frac{1}{2} \cos(2\pi f_c t_1 + 2\pi F \frac{i-j}{T_b} t_1 + \theta_{0,i} - \theta_{0,j}) \, dt_1$$

$$= \sum_{i=0}^{N-1} \frac{2}{T_b^2} NoEd_{0,i} \frac{1}{2} Tb = N \frac{No}{Tb} Ed_{0,i}^2 \tag{B.3}$$

Sesuai dengan pendekatan dan ciri-ciri yang digunakan

$$\int_{kTb}^{(k+1)Tb} \cos(4\pi f_c t_1 + 2\pi F \frac{i-j}{T_b} t_1 + \theta_{0,i} - \theta_{0,j}) \, dt_1 \cong 0 \tag{B.4}$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{N-1} \sum_{j=0}^{N-1} \int_{kTb}^{(k+1)Tb} \frac{1}{2} \cos(2\pi F \frac{i-j}{T_b} t_1 + \theta_{0,i} - \theta_{0,j}) \, dt_1 = 0$$

Pendekatan pada Persamaan (B-4), $\frac{2f_c + F(i+j)}{T_b} >> \frac{1}{T_b}$. Kondisi ini memuaskan bagi kondisi frekuensi *carrier* (dalam range GHz) dan *data rate* (pada orde Mbauds/sec). Pada Persamaan (A-5) hasil dari integral dari nilai integer perioda diantara gelombang *cosinus* dan perioda : $T_b / [F(i-j)]$ untuk $i \neq j$.

Untuk EGC pada kanal *fading* Rayleigh, *variance* dari *noise* diberikan sebagai berikut :

$$\sigma_\eta^2 = N \frac{N_0}{T_b} \tag{B.5}$$

Sifat stasistik interference

Sesuai ciri stastistik yang digunakan :

$$Ea_m[k] = \frac{1}{2}(-1) + \frac{1}{2}(1) = 0 \qquad Ea_m^2[k] = \frac{1}{2}(-1)^2 + \frac{1}{2}(1)^2 = 1$$

$$E \cos^2(\theta_{0,i} - \theta_{m,i}) = \int_0^{2\pi} \frac{1}{2\pi} \cos^2 \tilde{\theta}_{m,i} \, d\tilde{\theta}_{m,i} = \frac{1}{2} \tag{B.6}$$

Komponen *interference*, β_{int}, untuk EGC adalah :

$$\beta_{\text{int}} = \sum_{m=1}^{M-1} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} \hat{a}_k^m(t)\hat{c}_i^m(t)\,\hat{c}_i^0(t)\,\rho_{m,i}\cos\widetilde{\theta}_{m,i} \tag{B.7}$$

Karena komponen *in-phase* variabel acak Rayleigh, $\rho_{m,i}\cos\widetilde{\theta}_{m,i}$, adalah Gaussian dan $a_k^m(t)$, $c_i^m(t)$, $c_i^0(t) \in \{-1,1\}$, β_{int} terdiri dari penjumlahan dari $(M$-1)xNxL *iid* Gaussian random varibel . Jadi β_{int} adalah Gaussian dengan *mean* dan *variance* :

$$E\beta_{\text{int}} = \sum_{m=1}^{M-1} \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} E\left\{a_m[k]\rho_{i,l}^m c_m[i]c_0[i]\cos\widetilde{\theta}_{m,i}\right\} \tag{B.8}$$

$$= \sum_{m=1}^{M-1} Ea_m[k] \sum_{i=0}^{N-1} \sum_{l=0}^{L-1} c_m[i]c_0[i]E\rho_{i,l}^m E\cos\widetilde{\theta}_{m,i}$$

$$\sigma_{\beta_{\text{int}}}^2 = (M-1)NLE\left\{a_m[k]\rho_{m,i}c_m[i]c_0[i]\cos\widetilde{\theta}_{m,i}\right\}^2 \tag{B.9}$$

$$= (M-1)NL\,c_m^2[i]c_0^2[i]Ea_m^2[k]\rho_{m,i}^2\,E\cos^2\widetilde{\theta}_{m,i}$$

$$= \frac{1}{2}(M-1)NLE\rho_{m,i}^2 = (M-1)\overline{p}_m$$

Dari catatan ini benar bahwa *variance interference local-mean power* untuk Rayleigh *fading* diperoleh besarnya sama. $\overline{p}_{m,i} = \frac{1}{2}NLE\rho_{m,i}^2$

```
function utamaperubahanrake
clear all
close all
%%parameter inputan
jumuser=5;                  %jumlah user total
pjkode=16;                  %panjangkode hadamard yang digunakan
jumsubcar=3;                        %jumlah subcarrier tiap user
rake=[1 3 5];               %jumlah rake yang digunakan
N=round(8000/jumsubcar);    %panjang data
fc=900*10^6;                %frekuensi carrier
B=1.25*10^6;                %Bandwith subcarrier=1,25Mhz maka periode
                            bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                      %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;         %vektor time
loop=15*jumsubcar;            %jumlah loop monte carlo
iterasi=2;                  %jumlah iterasi

h=waitbar(0,'Sabarya nunggunya:).......');
i=1;
for i=1:iterasi
   waitbar(i/iterasi,h)

out1(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(1),N,fc,B,jum,t,loo
p)

out2(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(2),N,fc,B,jum,t,loo
p)

out3(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(3),N,fc,B,jum,t,loo
p)
end
```

```
berout(1,:)=sum(out1,1)/iterasi;
berout(2,:)=sum(out2,1)/iterasi;
berout(3,:)=sum(out3,1)/iterasi;


save perubahanrake3subcarrier
semilogy(berout(1,:),'r'),hold on,grid on
semilogy(berout(2,:),'b'),hold on,grid on
semilogy(berout(3,:),'g'),hold on,grid on
close(h)
legend('No RAKE','3 RAKE','5 RAKE');
title('Grafik Perubahan Rake');
xlabel('SNR');
ylabel('BER');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
clear all
close all
%%parameter inputan
jumuser=5;                    %jumlah user total
pjkode=16;                    %panjangkode hadamard yang digunakan
jumsubcar=5;                          %jumlah subcarrier tiap user
rake=[1 3 5];                 %jumlah rake yang digunakan
N=round(8000/jumsubcar);    %panjang data
fc=900*10^6;                  %frekuensi carrier
B=1.25*10^6;                  %Bandwith subcarrier=1,25Mhz maka periode
                              bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                        %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;           %vektor time
loop=15*jumsubcar;              %jumlah loop monte carlo
iterasi=2;                    %jumlah iterasi
```

```
h=waitbar(0,'Sabarya nunggunya:).......');
i=1;
for i=1:iterasi
   waitbar(i/iterasi,h)

out1(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(1),N,fc,B,jum,t,loo
p)

out2(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(2),N,fc,B,jum,t,loo
p)

out3(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(3),N,fc,B,jum,t,loo
p)
end
berout(1,:)=sum(out1,1)/iterasi;
berout(2,:)=sum(out2,1)/iterasi;
berout(3,:)=sum(out3,1)/iterasi;
save perubahanrake5subcarrier
semilogy(berout(1,:),'r'),hold on,grid on
semilogy(berout(2,:),'b'),hold on,grid on
semilogy(berout(3,:),'g'),hold on,grid on
close(h)
legend('No RAKE','3 RAKE','5 RAKE');
title('Grafik Perubahan Rake');
xlabel('SNR');
ylabel('BER');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
clear all
close all
%%parameter inputan
jumuser=5;                    %jumlah user total
```

```
pjkode=16;                      %panjangkode hadamard yang digunakan
jumsubcar=7;                            %jumlah subcarrier tiap user
rake=[1 3 5];                   %jumlah rake yang digunakan
N=round(8000/jumsubcar);   %panjang data
fc=900*10^6;                    %frekuensi carrier
B=1.25*10^6;                    %Bandwith subcarrier=1,25Mhz maka periode
                                bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                          %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;             %vektor time
loop=15*jumsubcar;                %jumlah loop monte carlo
iterasi=2;                      %jumlah iterasi


h=waitbar(0,'Sabarya nunggunya:).......');
i=1;
for i=1:iterasi
   waitbar(i/iterasi,h)

out1(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(1),N,fc,B,jum,t,loo
p)

out2(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(2),N,fc,B,jum,t,loo
p)

out3(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(3),N,fc,B,jum,t,loo
p)
end
berout(1,:)=sum(out1,1)/iterasi;
berout(2,:)=sum(out2,1)/iterasi;
berout(3,:)=sum(out3,1)/iterasi;
save perubahanrake7subcarrier
semilogy(berout(1,:),'r'),hold on,grid on
```

```
semilogy(berout(2,:),'b'),hold on,grid on
semilogy(berout(3,:),'g'),hold on,grid on
close(h)
legend('No RAKE','3 RAKE','5 RAKE');
title('Grafik Perubahan Rake');
xlabel('SNR');
ylabel('BER');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%
clear all
close all
%%parameter inputan
jumuser=5;                  %jumlah user total
pjkode=16;                  %panjangkode hadamard yang digunakan
jumsubcar=9;                        %jumlah subcarrier tiap user
rake=[1 3 5];               %jumlah rake yang digunakan
N=round(8000/jumsubcar);    %panjang data
fc=900*10^6;                %frekuensi carrier
B=1.25*10^6;                %Bandwith subcarrier=1,25Mhz maka periode
                            bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                      %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;         %vektor time
loop=15*jumsubcar;            %jumlah loop monte carlo
iterasi=2;                  %jumlah iterasi

h=waitbar(0,'Sabarya nunggunya:).......');
i=1;
for i=1:iterasi
   waitbar(i/iterasi,h)

out1(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(1),N,fc,B,jum,t,loo
p)
```

```
out2(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(2),N,fc,B,jum,t,loo
p)

out3(i,:)=perubahanrake(jumuser,pjkode,jumsubcar,rake(3),N,fc,B,jum,t,loo
p)
end
berout(1,:)=sum(out1,1)/iterasi;
berout(2,:)=sum(out2,1)/iterasi;
berout(3,:)=sum(out3,1)/iterasi;
save perubahanrake9subcarrier
semilogy(berout(1,:),'r'),hold on,grid on
semilogy(berout(2,:),'b'),hold on,grid on
semilogy(berout(3,:),'g'),hold on,grid on
close(h)
legend('No RAKE','3 RAKE','5 RAKE');
title('Grafik Perubahan Rake');
xlabel('SNR');
ylabel('BER');
```

perubahanrake.m

```
function perubahanrake=perubahanrake(jumuser,pjkode,jumsubcar,rake,
                                      N,fc,B,jum,t,loop)


%clear all
%global jum f t M dt gdb gn jumuser
% %%parameter inputan
% jumuser=2;                  %jumlah user total
% pjkode=16;                  %panjangkode hadamar yang digunakan
% jumsubcar=3;                %jumlah subcarrier tiap user
```

```
% rake=1;                    %jumlah rake yang digunakan
% N=1000;                              %panjang data
% fc=900*10^6;               %frekuensi carrier
% B=1.25*10^6;               %Bandwith subcarrier=1,25Mhz maka
                             periode bit 0.8 microsecd,B=1/Tc;Tc=0.8
                             microsecd
% jum=2;                     %jumlah bit pulseshaping pembentuk
pulsa
% t=0:jum*N*pjkode-1;        %vektor time
% loop=3;                    %jumlah loop monte carlo

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% loping monte carlo
tic
for lp=1:loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%pengulangan pengiriman sebanyak jumlah user
   for m=1:jumuser
      %urutan kode hadamard yang digunakan oleh user
      esc=m+1;
      %proses pemanggilan function txcdmarake
[data(m,:),Rx(m,:),kode(:,:,m),f]=txmccdmarake(N,pjkode,esc,jumsubcar,m,
                                          fc,B,jum,t);
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%   %sinyal noise dan multipath fading
%%   %proses multpath fading: sinyal ditunda sebanyak 6 tap,sesuai
standar ieee
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%   % Proses multipath fading masing-masing user
   for m=1:jumuser
      clear chan
      chan=rayleighchan(tc,fd,tau,gdb);
      for k=1:path
         % kanal Rayleigh fading dengan delay.
         ydelay(k,:,m)=[zeros(1,dt(k)) Rx(m,1:end-dt(k))];
         ymulfad(k,:,m)=filter(chan,ydelay(k,:,m));
       end
   end
   %% interference oleh path yang lain pada user satu
   yintipath=sum(ymulfad(:,:,1),1);
   %% interference oleh path pada user yang lain
   yintmulfad=sum(ymulfad(:,:,2:end),1);
   %% interference oleh user yang lain
   yinterfrc=sum(yintmulfad(:,:,2:end),3);
   %%sinyal dengan pengaruh interference
   yrx=yintipath+0.1.*yinterfrc;
   axix=1;
   for SNR=1:2:30
      %% penambahan noise
      clear yreceive
      yreceive=awgn(yrx,SNR,0);
      %yreceive=awgn(yrx,SNR,'measured');
      %figure;plot(real(yreceive));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% %      % proses dipenerima user 1 sebagai acuan.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
jumbiterr(loop,axix)=rxmccdmarake(data(1,:),yreceive,N,jumsubcar,kode(:,:
,1),
                                        rake,jum,f,dt,t,pjkode);
      %% keluaran dalam bentuk jumlah bit yang error
      axix=axix+1;
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
% save jumbiterr jumbiter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
end
toc
%jumlah error bit yg terjadi terhadap total data yang dikirim
Bittotsalah=sum(jumbiterr,1);
BERR=Bittotsalah/N/loop;
perubahanrake=BERR;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
```

txmccdmarake.m

```
function [data,ytransmit,kode,f]=txmccdmarake(N,pjkode,esc,jumsubcar,m,
                                                    fc,B,jum,t)

%N=2;
%SNR=20;
%data=[1 -1]
%pjkode=panjang kode hadamard yang digunakan
%jumsubcar=jumlah subcarrier
%parameter modulasi frekuensi yang digunakan
%Bandwith subcarrier=1,25Mhz==>maka Tc=5 microsecond
%fc=900*10^6;
```

```
%B=1/Tc;Tc=0.8 microsecd
%B=1.25*10^6;
% periode bit 0.8 microsecd untuk bandwidth 1,25Mh
%jum=1;
%t=0:jum*N*pjkode-1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

%generate data modulasi BPSK
data=2*randint(1,N,[0 1])-1;
%generate kode yang digunakan
kode=kodetunda(pjkode,esc,jumsubcar);

yspread=zeros(jumsubcar,length(data)*pjkode);
yifft=zeros(jumsubcar,length(data)*pjkode);
ypulse=zeros(jumsubcar,length(data)*pjkode*jum);
ymod=zeros(jumsubcar,length(data)*pjkode*jum);

%proses spreading
yspread=kron(data,kode);

% proses ifft
 for i=1:jumsubcar
   for k=1:N
       yifft(i,pjkode*k-(pjkode-1):pjkode*k)=ifft(yspread(i,pjkode*k-(pjkode
                                                -1):pjkode*k),pjkode);
   end

% proses pulseshaping
   ypulse(i,:)=rectpulse(yifft(i,:),jum);

% proses modulasi
```

```
   f(i)=(fc+B*(4*i-3)/2);
   ymod(i,:)=ypulse(i,:).*cos(2*pi/jum*f(i).*t);
 end
```

```
%proses pengiriman asinkron user
if m==1
   asink=0;
else
   asink=randsrc(1,1,[1:5]);
end
yasin=[zeros(jumsubcar,asink) ymod(:,1:length(ymod)-asink)];

%siyal yang dikirimkan
ytransmit=ParToSer(yasin);
```

rxmccdmarake.m

```
function bitsalah=rxmccdmarake(data,yreceive,N,jumsubcar,kode,rake,
                              jum,f,dt,t,pjkode)

%siyal terima
ypath=SerToPar(yreceive,pjkode*N*jum);

ydmod=zeros(jumsubcar,length(ypath));
ydepulse=zeros(rake,length(data)*pjkode,jumsubcar);
yfft=zeros(rake,length(data)*pjkode,jumsubcar);
ydespread=zeros(rake,length(data)*pjkode,jumsubcar);
```

```
yrake=zeros(rake,length(data),jumsubcar);

% proses demodulasi sinyal
for i=1:jumsubcar
   ydmod(i,:)=ypath(i,:).*cos(2*pi/jum*f(i).*t);




% proses depulse
   for k=1:rake
      ydepulse(k,:,i)=intdump(ydmod(i,:),jum);

   % proses fft
     for h=1:N
        yfft(k,pjkode*h-(pjkode-1):pjkode*h,i)=fft(ydepulse(k,pjkode*h-
(pjkode
                                                            -1):pjkode*h,i),pjkode);
     end

   % proses despreading dengan metode EGC tiap rake
     bn=1;      % EGC
     delaykode(k,:,i)=bn*[zeros(1,dt(k)) kode(i,1:length(kode)-dt(k))];
     ydespread(k,:,i)=despreading(yfft(k,:,i),pjkode,delaykode(k,:,i));

   % proses integral
      for h=1:length(data)
         yrake(k,h,i)=integral(ydespread(k,pjkode*h-(pjkode-1):pjkode*h,i));
      end

   end
end
```

```
%Proses decision tiap jumsubcar
if rake==1
   for i=1:jumsubcar
       ysubcar(i,:)=yrake(:,:,i);
   end
else
   for i=1:jumsubcar
       ysubcar(i,:)=sum(yrake(:,:,i));
   end
end

%   keluaran akhir masing subcarrier
yakhir=limiter(sum(ysubcar));
%data
%keluaran dalam jumlah bit yang salah
bitsalah=symerr(yakhir,data);
```

kodetunda.m

```
function kode=kodetunda(pjkode,urutkode,jumsubcar)

%  pjkode=4;
%  urutkode=2;
%  jumsubcar=3;

usedkode=hadamard(pjkode);
Kodeasli=usedkode(urutkode,:);

for i=1:jumsubcar;
   if i==1
       kode(i,:)= Kodeasli;
```

```
    else
        kode(i,:)=[Kodeasli(pjkode-i+2:pjkode) ...
                   Kodeasli(1:pjkode-i+1)];
    end
end
```

limiter.m

```
function y=limiter(x)
yreal=ones(1,length(x));
for i=1:length(x)
    if real(x(i))>=0
        yreal(i)=1;
    else
        yreal(i)=-1;
    end
end
y=yreal;
```

ParToSer.m

```
function L=ParToSer(masukan)
% D=data masukan
% masukan=[1 2 3;4 5 6;7 8 9]
K=masukan;

[p q]=size(K);
```

```
b=zeros(1,p*q);
for r =1:p
        b(1,q*(r)-q+1:q*(r))=K(r,:);
end
L=b;
```

SerToPar.m

```
function DataPar=SerToPar(b,jumsubcar);
% memberikan inputan
% b=[1 2 3 4 5 6 7 8 9]
% jumsubcar=3;
% m=jumlah subcarrier
% b=(1:16)
% m=4;

D=zeros(length(b)/jumsubcar,jumsubcar);
[p q]=size(D);

for r=1:p
    D(r,:)=b(1,q*(r)-q+1:q*(r));
end
DataPar=D;
```

Utamaperubahanuser.m

```
function utamaperubahanuser
```

```
clear all
%%parameter inputan
jumuser=15;                    %jumlah user total
pjkode=16;                     %panjangkode hadamard yang digunakan
jumsubcar=3;                   %jumlah subcarrier tiap user
rake=[1 3 5];                  %jumlah rake yang digunakan
N=round(1000/jumsubcar);    %panjang data
fc=900*10^6;                   %frekuensi carrier
B=1.25*10^6;                   %Bandwith subcarrier=1,25Mhz maka periode
                               bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                         %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;            %vektor time
loop=1*jumsubcar;               %jumlah loop monte carlo
SNR=18;                        %penguatan dalam dB ketika 3 subcarrier
                               mencapai ber 1-e5
iterasi=1;                     % jumlah iterasi yang dilakukan

h=waitbar(0,'Tunggu yang Sabarya:).......');
i=1;
for i=1:iterasi
   waitbar(i/iterasi,h)
out1(i,:)=perubahanuser(jumuser,pjkode,jumsubcar,rake(1),N,fc,B,jum,t,
                        loop,SNR)
 out2(i,:)=perubahanuser(jumuser,pjkode,jumsubcar,rake(2),N,fc,B,jum,t,
                        loop,SNR)
 out3(i,:)=perubahanuser(jumuser,pjkode,jumsubcar,rake(3),N,fc,B,jum,t,
                        loop,SNR)
end
beroutrake(1,:)=sum(out1,1)/iterasi;
beroutrake(2,:)=sum(out2,1)/iterasi;
beroutrake(3,:)=sum(out3,1)/iterasi;
```

```
save perubahan user
close(h)
semilogy(beroutrake(1,:),'r'),hold on,grid on
semilogy(beroutrake(2,:),'b'),hold on,grid on
semilogy(beroutrake(3,:),'g'),hold on,grid on
legend('No RAKE','3 RAKE','5 RAKE');
title('Grafik Perubahan jumlah user dengan SNR=18');
xlabel('jumlah user');
ylabel('BER');
```

perubahanuser.m

```
function perubahanuser=perubahanuser(jumuser,pjkode,jumsubcar,rake,N,
                                     fc,B,jum,t,loop,SNR)

% clear all
% global jum f t M dt gdb gn jumuser
% %%parameter inputan
% jumuser=10;            %jumlah user total
% pjkode=16;             %panjangkode hadamar yang digunakan
% jumsubcar=3;           %jumlah subcarrier tiap user
% rake=1;                %jumlah rake yang digunakan
% N=1000;                    %panjang data
% fc=900*10^6;           %frekuensi carrier
% B=1.25*10^6;           %Bandwith subcarrier=1,25Mhz maka periode
                         bit 0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
% jum=2;                 %jumlah bit pulseshaping pembentuk pulsa
% t=0:jum*N*pjkode-1;    %vektor time
% loop=1;                %jumlah loop monte carlo
% SNR=10;                %penguatan dalam dB

axix=0;
for user=1:5:jumuser
```

```
   axix=axix+1;

% loping monte carlo
   tic
   for lp=1:loop
      lp




%pengulangan pengiriman sebanyak jumlah user
      for m=1:user
%urutan kode hadamard yang digunakan oleh user
       esc=m+1;
%proses pemanggilan function txcdma

[data(m,:),Rx(m,:),kode(:,:,m),f]=txmccdmauser(N,pjkode,esc,jumsubcar,m,
                                                 fc,B,jum,t);
      end
tau=[0e-7 8e-7 16e-7 24e-7 32e-7 40e-7]*jum;
gdb=[0 -1 -9 -10 -15 -20];       %redaman masing-masing path dlm dB
gn=10.^(gdb/10);                  %redaman masing-masing path dlm decimal
tc=8e-7/jum;                      %periode chip yang masuk kekanal dlm s
fd=10;                            %frekuansi doppler dlm hz
dt=[0 1 2 3 4 5]*jum;             %delay path dalam catuan chip,
dt=round(tau./tc.*jum);
path=length(dt);                  %jumlah path

%kondisi kanal yang pengaruh mempengaruhi sinyal
%sinyal noise dan multipath fading
%proses multpath fading: sinyal ditunda sebanyak 6 tap,sesuai standar ieee
% Proses multipath fading masing user
```

```
        for m=1:user
            for k=1:path
            % Create Rayleigh fading channel object with delay.
                clear chan
                chan=rayleighchan(tc,fd,tau,gdb);
                ydelay(k,:,m)=[zeros(1,dt(k)) Rx(m,1:end-dt(k))];
                ymulfad(k,:,m)=filter(chan,ydelay(k,:,m));
             end
        end

% interference oleh path yang lain pada user satu
        yintipath=sum(ymulfad(:,:,1),1);
% interference oleh path pada user yang lain
        yintmulfad=sum(ymulfad(:,:,2:end),1);
% interference oleh user yang lain
        yinterfrc=sum(yintmulfad(:,:,2:end),3);
%sinyal dengan pengaruh interference
        yrx=yintipath+yinterfrc;

% penambahan noise
        clear yreceive
        %yreceive=awgn(yrx,SNR,0);
        yreceive=awgn(yrx,SNR,'measured');
        %figure;plot(real(yreceive));

% proses dipenerima user 1 sebagai acuan.
jumbiterr(loop,axix)=rxmccdmauser(data(1,:),yreceive,N,jumsubcar,kode(:,:,
1),
                                                            rake,jum,f,dt,t,pjkode);
% keluaran dalam bentuk jumlah bit yang error
    end
end
```

```
toc
%jumlah error bit yg terjadi terhadap total data yang dikirim
Bittotsalah=sum(jumbiterr,1);
BERR=Bittotsalah/N/loop;
perubahanuser=BERR;
```

tcmccdmarakeuser.m

```
function [data,ytransmit,kode,f]=txmccdmauser(N,pjkode,esc,jumsubcar,m,
                                                fc,B,jum,t)

%N=2;
%SNR=20;
%data=[1 -1]
%panjang kode hadamard yang digunakan
%pjkode=pjkode;
%jumlah subcarrier
%jumsubcar=jumsubcar;
%parameter modulasi frekuensi yang digunakan
%Bandwith subcarrier=1,25Mhz==>maka Tc=5 microsecond
%fc=900*10^6;
%B=1/Tc;Tc=0.8 microsecd
%B=1.25*10^6;
% periode bit 0.8 microsecd untuk bw 1,25Mh
%jum=1;
%t=0:jum*N*pjkode-1;
```

```
%generate data modulasi BPSK
data=2*randint(1,N,[0 1])-1;
%generate kode yang digunakan
kode=kodetunda(pjkode,esc,jumsubcar);

yspread=zeros(jumsubcar,length(data)*pjkode);
yifft=zeros(jumsubcar,length(data)*pjkode);
ypulse=zeros(jumsubcar,length(data)*pjkode*jum);
ymod=zeros(jumsubcar,length(data)*pjkode*jum);


%proses spreading
yspread=kron(data,kode);

% proses ifft
 for i=1:jumsubcar
    for k=1:N
        yifft(i,pjkode*k-(pjkode-1):pjkode*k)=ifft(yspread(i,pjkode*k-(pjkode
                                                -1):pjkode*k),pjkode);
    end

% proses pulseshaping
    ypulse(i,:)=rectpulse(yifft(i,:),jum);

% proses modulasi
    f(i)=(fc+B*(4*i-3)/2);
    ymod(i,:)=ypulse(i,:).*cos(2*pi/jum*f(i).*t);
 end

%proses pengiriman asinkron user
if m==1
    asink=0;
```

```
else
   asink=randsrc(1,1,[1:5]);
end
yasin=[zeros(jumsubcar,asink) ymod(:,1:length(ymod)-asink)];

%siyal yang dikirimkan
ytransmit=ParToSer(yasin);
```

rxmccdmauser.m

```
function bitsalah=rxmccdmauser(data,yreceive,N,jumsubcar,kode,rake,jum,
                                           f,dt,t,pjkode)

[baris,kolom]=size(kode);
%%siyal terima
ypath=SerToPar(yreceive,kolom*N*jum);

ydmod=zeros(jumsubcar,length(ypath));
ydepulse=zeros(rake,length(data)*kolom,jumsubcar);
yfft=zeros(rake,length(data)*kolom,jumsubcar);
ydespread=zeros(rake,length(data)*kolom,jumsubcar);
yrake=zeros(rake,length(data),jumsubcar);

% proses demodulasi sinyal
for i=1:jumsubcar
   ydmod(i,:)=ypath(i,:).*cos(2*pi/jum*f(i).*t);

% proses depulse
   for k=1:rake
```

```
        ydepulse(k,:,i)=intdump(ydmod(i,:),jum);

% proses fft
       for h=1:N
          yfft(k,kolom*h-(kolom-1):kolom*h,i)=fft(ydepulse(k,kolom*h-(kolom
                                                          -1):kolom*h,i),kolom);
       end

% proses despreading dengan metode EGC tiap rake
       bn=1; % EGC
       delaykode(k,:,i)=bn*[zeros(1,dt(k)) kode(i,1:length(kode)-dt(k))];
       ydespread(k,:,i)=despreading(yfft(k,:,i),kolom,delaykode(k,:,i));

% proses integral
        for h=1:length(data)
           yrake(k,h,i)=integral(ydespread(k,kolom*h-(kolom-1):kolom*h,i));
        end

    end
end

%Proses decision tiap jumsubcar
if rake==1
    for i=1:jumsubcar
        ysubcar(i,:)=yrake(:,:,i);
    end
else
    for i=1:jumsubcar
        ysubcar(i,:)=sum(yrake(:,:,i));
    end
end
```

```
%   keluaran akhir masing subcarrier
yakhir=limiter(sum(ysubcar));
%data

%keluaran dalam jumlah bit yang salah
bitsalah=symerr(yakhir,data);
```

kodetunda.m

```
function kode=kodetunda(pjkode,urutkode,jumsubcar)

%  pjkode=4;
%  urutkode=2;
%  jumsubcar=3;

usedkode=hadamard(pjkode);
Kodeasli=usedkode(urutkode,:);

for i=1:jumsubcar;
   if i==1
      kode(i,:)= Kodeasli;
   else
      kode(i,:)=[Kodeasli(pjkode-i+2:pjkode) ...
                 Kodeasli(1:pjkode-i+1)];
   end
end
```

limiter.m

```
function y=limiter(x)
yreal=ones(1,length(x));
for i=1:length(x)
   if real(x(i))>=0
      yreal(i)=1;
   else
      yreal(i)=-1;
   end
end
y=yreal;
```

ParToSer.m

```
function L=ParToSer(masukan)
% D=data masukan
% masukan=[1 2 3;4 5 6;7 8 9]
K=masukan;

[p q]=size(K);
b=zeros(1,p*q);
for r =1:p
        b(1,q*(r)-q+1:q*(r))=K(r,:);
end
L=b;
```

SerToPar.m

```
function DataPar=SerToPar(b,jumsubcar);
% memberikan inputan
```

```
% b=[1 2 3 4 5 6 7 8 9]
% jumsubcar=3;
% m=jumlah subcarrier
% b=(1:16)
% m=4;

D=zeros(length(b)/jumsubcar,jumsubcar);
[p q]=size(D);



for r=1:p
    D(r,:)=b(1,q*(r)-q+1:q*(r));
end
DataPar=D;
```

VERYFIKASI KANAL

Kanal AWGN

Proses *Gaussian*, n(t), adalah sebuah fungsi random yang memiliki nilai, n, dalam setiap waktu yang berubah-ubah, t, adalah bersifat statistik yang memenuhi karakter dari *probability density function (pdf) Gaussian*, p(n):

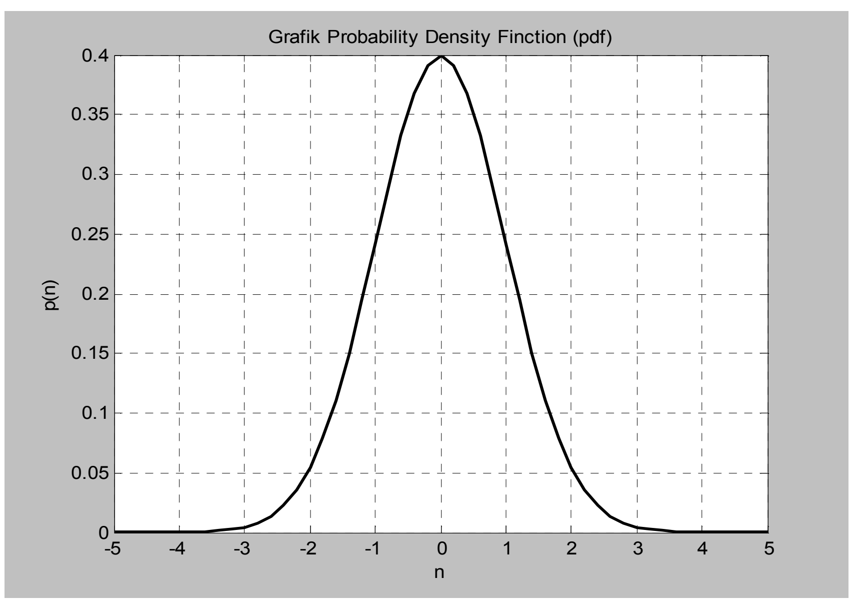
$$p(n) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{n}{\sigma}\right)^2\right] \quad \text{(C.1)}$$

dimana σ^2 adalah variansi dari n. *Gaussian density function* ternormalisasi pada sebuah proses *zero-mean* dihasilkan dengan menganggap bahwa $\sigma = 1$. Nilai p(n) maksimum pada saat n = 0, yaitu $\frac{1}{\sqrt{2\pi}}$

Berikut ini skrip untuk melukiskan persamaan C.1, yaitu:

```
a=-5:0.2:5;
b=(1/sqrt(2*pi))*2.7183.^(-a.^2/2);
plot(a,b,'k-')
```

Gambar yang dihasilkan dari skrip diatas adalah sebagai berikut :



Gambar D.1 AWGN Teory

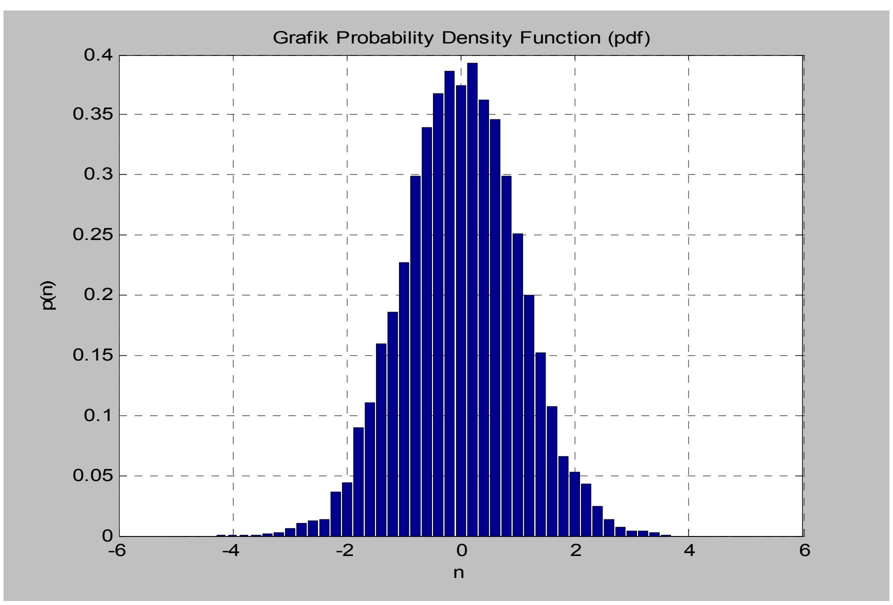
Pembangkitan *noise* AWGN dilakukan dengan perintah awgn yang disediakan oleh MATLAB. Verifikasi kanal *AWGN* dilakukan dengan skrip perintah sebagai berikut :

```
a=-5:0.2:5;
c=zeros(1,10000);
d=awgn(c,0,0);
e=hist(d,a)/(0.2*10000);
bar(a,e)
v = var(d)
m = mean(d)
mx = max(e)
```

Data yang dihasilkan dari skrip diatas adalah sebagai berikut :

```
v = 1.0116
m = 0.026443
mx= 0.395
```

Gambar yang dihasilkan dari skrip diatas adalah sebagai berikut :



Gambar D.2 AWGN Simulasi

Dari skrip diatas diperoleh data variansi (v) = 1,0116 $\approx$ 1, rata-rata (m) = 0,0026443 $\approx$ 0 dan nilai maksimal(mx) = 0,395 $\approx \frac{1}{\sqrt{2\pi}}$.

Fading Rayleigh

Fading Rayleigh kanal dibangkitkan dengan menggunakan *tool* pada MATLAB dengan perintah *rayleigchan.* Tool ini memerlukan inputan periode simbol, frekuensi Doppler, *delay path,* dan *gain path* dimana cara kerja dari toolbox model *fading* kanal ini berfungsi sebagai linear filter FIR, dengan memberikan pembebanan pada masing-masing *tap*:

$$g_n = \sum_k \sin c(\tau_k / T - n) h_k \qquad for - N_1 \le n \le N_2$$

dimana :

- Setiap path utama memiliki satu persamaan.
- $\{\tau_k\}$merupakan nilai dari *delay path.*
- T merupakan periode simbol.
- N_1 dan N_2 merupakan parameter yang dipilih dimana $|g_n|$ berharga kecil pada saat n kecil dari $-N_1$ dan besar dari N_2.
- N_1 merupakan harga dari object ChannelFilterDelay property.
- $\{h_k\}$merupakan komplex *gain path* dimana tidak saling berhubungan satu dengan yang lain.

Parameter h_k merupakan *particular gain path* yang dibangkitkan dengan tahapan sebagai berikut:

1. Tahap pertama akan digenerate White Goussian Noise
2. Kemudian Noise ini dilewatkan melalui filter yang memiliki respon power spektrum sesuai Jake Doppler spektrum.
3. Tahap berikutnya nilai-nilai tersebut disisipkan sedemikian sehingga didapatkan sample periode yang kosisten dengan sinyal.
4. Kemudian dilakukan penyesuaian untuk memperoleh nilai yang benar dari rata-rata *path gain*.

Secara umum cara kerja dari toolbox menggunakan pemodelan kanal *fading* Jake's Model

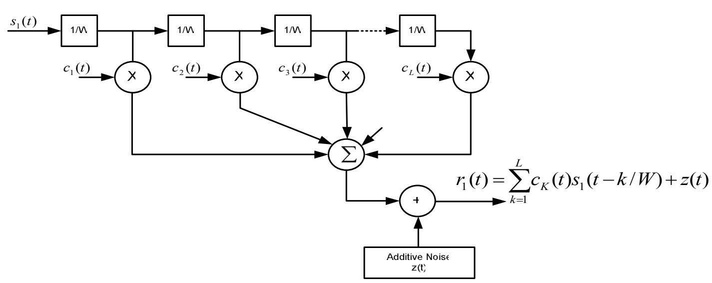
Kanal *Multipath Fading*

Respon kanal *multipath rayleigh fading* dikarakterisasi dengan pdf berikut:

$$p(r) = \begin{cases} \dfrac{r}{\sigma^2} e^{\left[-\frac{r^2}{2\sigma^2}\right]} & \text{, untuk } r \geq 0 \\ 0 & \text{, untuk r lainnya} \end{cases} \qquad (C.2)$$

dimana *r* adalah amplituda selubung sinyal yang diterima, σ adalah nilai tegangan rms dari sinyal diterima sebelum detektor selubung dan σ^2 adalah daya rata-rata dari sinyal diterima sebelum detektor selubung.

Kanal *frequency-selective fading* dapat dimodelkan sebagai *tapped delay line* dengan interval 1/W dan koefisen tap c_n (t) sebagai berikut :



Gambar D.3 Model *tapped delay line* kanal *frequency selective fading*

Koefisien tap $c_n(t)$ berubah waktu yang merupakan proses komplek acak tetap. Jika kanal dengan *fading Rayleigh*, maka nilai magnitude $|c_n(t)| \equiv \alpha_n(t)$ terdistribusi *Rayleigh* dan fase $\Phi_n(t)$ terdistribusi *uniform*.

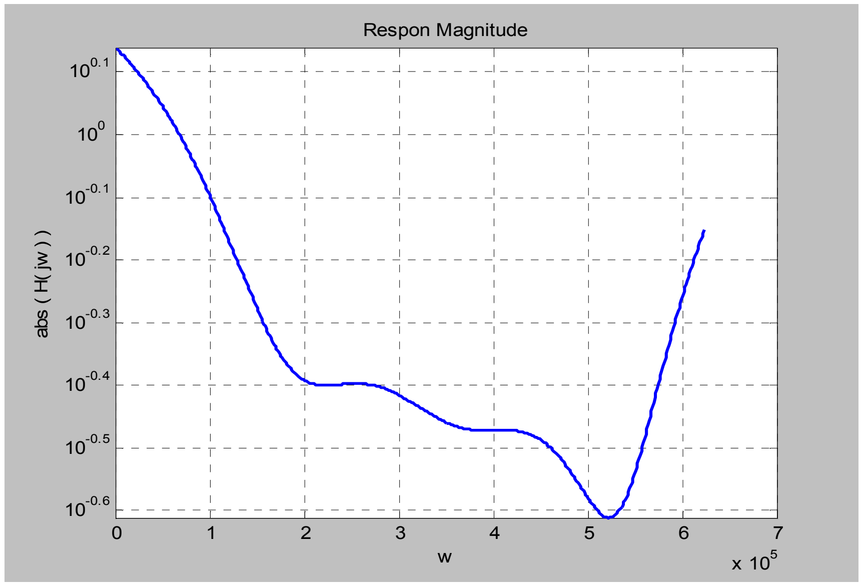
Pada simulasi digunakan model 6 *tap* berdasarkan IMT2000.. Model ini menggunakan 6 *ray-Rayleigh fading* yang independent seperti pada Gambar 3.3. Dimana besarnya dari Rayleigh fading simulator menggunakan fungsi Matlab, dibangkitkan dengan skrip dibawah ini :

```
%Parameter kanal;
        tau=[0 1e-7 10e-7 20e-7 30e-7 40e-7];
```

```
gdb=[0 -1 -9 -10 -15 -20];
fd=6.15;
bit=1250000;
%Pemanggilan nilai nilai dari kanal masing-masing rayleigh fading simulator
%untuk 6 Tap.
chan = rayleighchan(1/bit,fd,tau,gdb);
%menampilakan Mode kanal yang digunakan
chan.ChannelType
%Numerator
%c.pathGains merupakan nilai-nilai tiap path yang dihasilkan masing- masing
%simulator fading
b=[chan.PathGains]
%Denumerator
a=1
%mendapatkan parameter fasa dan amplitudo dalam domain frekuesi
[h,w]=freqz(b,a,512,bit);
mag=abs(h);
phase=angle(h)*180/pi;
%menampilkan
semilogy(w,mag);grid on
title('Frequency = 0 Hz')
title('Respon Magnitude')
xlabel('w');
ylabel('abs ( H( jw ) )');
```

Keluaran yang dihasilkan dari skrip diatas adalah sebagai berikut :

Rayleigh



Gambar D.4 Respon *Frekuensi Selective*

%===

% VALIDASI DATA 1 USER, 3 SUBCARRIER, 4 RAKE

%===

Multiuserrake4.m

```
function multiuserrake4=multiuserrake4(jumuser,pjkode,jumsubcar,rake,N,loop)


clear all
global jum f t M dt gdb gn jumuser
%%parameter inputan
jumuser=1;                  %jumlah user total
pjkode=16;                  %panjangkode hadamar yang digunakan
jumsubcar=3;                %jumlah subcarrier tiap user
rake=1;                     %jumlah rake yang digunakan
totalbit=1000000;           %total data yang dikirimkan
loop=1;%*jumsubcar;         %jumlah loop monte carlo
N=round(3000/jumsubcar);    %panjang data dalam setiap pengiriman
fc=2*10^9;                  %frekuensi carrier
```

```
B=1.25*10^6;                    %Bandwith subcarrier=1,25Mhz maka periode bit
                                0.8 microsecd,B=1/Tc;Tc=0.8 microsecd
jum=2;                          %jumlah bit pulseshaping pembentuk pulsa
t=0:jum*N*pjkode-1;             %vektor time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% loping monte carlo
tic
for lp=1:loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%pengulangan pengiriman sebanyak jumlah user
   for m=1:jumuser
 %urutan kode hadamard yang digunakan oleh user
      esc=m+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%proses pemanggilan function txcdma4
[data(m,:),Rx(m,:),kode(:,:,m),f]=txmccdma4(N,pjkode,esc,jumsubcar,m,
                                                  fc,B,jum,t);
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

txmccdma4.m

```
function [data,ytransmit,kode,f]=txmccdma4(N,pjkode,esc,jumsubcar,
                                                  m,fc,B,jum,t)


%N=2;
%SNR=20;
%data=[1 -1]
%panjang kode hadamard yang digunakan
%pjkode=pjkode;
%jumlah subcarrier
%jumsubcar=jumsubcar;
%parameter modulasi frekuensi yang digunakan
%Bandwith subcarrier=1,25Mhz==>maka Tc=5 microsecond
```

```
%fc=900*10^6;
%B=1/Tc;Tc=0.8 microsecd
%B=1.25*10^6;
% periode bit 0.8 microsecd untuk bw 1,25Mh
%jum=1;
%t=0:jum*N*pjkode-1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%generate data modulasi BPSK
data=2*randint(1,N,[0 1])-1;
%generate kode yang digunakan
kode=kodetunda(pjkode,esc,jumsubcar);

yspread=zeros(jumsubcar,length(data)*pjkode);
yifft=zeros(jumsubcar,length(data)*pjkode);
ypulse=zeros(jumsubcar,length(data)*pjkode*jum);
ymod=zeros(jumsubcar,length(data)*pjkode*jum);

%proses spreading
yspread=kron(data,kode);
%      yspread(i,:)=spreading(data,pjkode,kode(i,:));

% proses ifft
  for i=1:jumsubcar
    for k=1:N
        yifft(i,pjkode*k-(pjkode-1):pjkode*k)=ifft(yspread(i,pjkode*k-(pjkode
                                                        -1):pjkode*k),pjkode);
    end

% proses pulseshaping
    ypulse(i,:)=rectpulse(yifft(i,:),jum);
```

```
% proses modulasi
    f(i)=(fc+B*(4*i-3)/2);
    ymod(i,:)=ypulse(i,:).*cos(2*pi/jum*f(i).*t);
 end

%proses pengiriman asinkron user
% dimana delay antara user, 0<delay<SF(Spreadingfaktor)
if m==1
    asink=0;
else
    asink=randsrc(1,1,[1:pjkode]);
end
yasin=[zeros(jumsubcar,asink) ymod(:,1:length(ymod)-asink)];

%siyal yang dikirimkan
ytransmit=ParToSer(yasin);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parameter Kanal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% tm berdasarkan imt2000 6 tap(setelah perhitungan)tm=0.1 microsecd
%% tc=5 microsecond untuk bw sinyal 1,25 Mhz
%% delay=[0 3.1 7.1 10.9 17.3 25.1]*1e-7; %dlm s
%% delaypath=[0 3.1 10.2 21.1 38.4 63.1]*1e-7; %dlm s
%% menggunalan function rayleughchan; chan = rayleighchan(ts,fd,tau,pdb)
%% ts=periode simbol dlm second,fd=frek doppler dalam hz,tau=delay
tau=[0e-7 8e-7 16e-7 24e-7 32e-7 40e-7]*jum;        %pembulatan sesuai dgn tc
gdb=[0 -1 -9 -10 -15 -20];        %redaman masing-masing path dlm dB
gn=10.^(gdb/10);                  %redaman masing-masing path dlm decimal
tc=8e-7/jum;                       %periode simbol dlm s
fd=6.25;                            %frekuansi doppler lingkungan normal dlm hz
dt=round(tau./tc.*jum);             %delay path dalam catuan chip,
dt=[0 2 4 6 8 10];
```

```
path=length(dt);                    %jumlah path
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%kondisi kanal yang pengaruh mempengaruhi sinyal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%prose pemanggilan function rxmccdmarake4
jumbiterr(loop,:)=rxmccdmarake4(data(1,:),yreceive,N,jumsubcar,kode(:,:,1),
                                        rake,jum,f,dt,t,pjkode);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

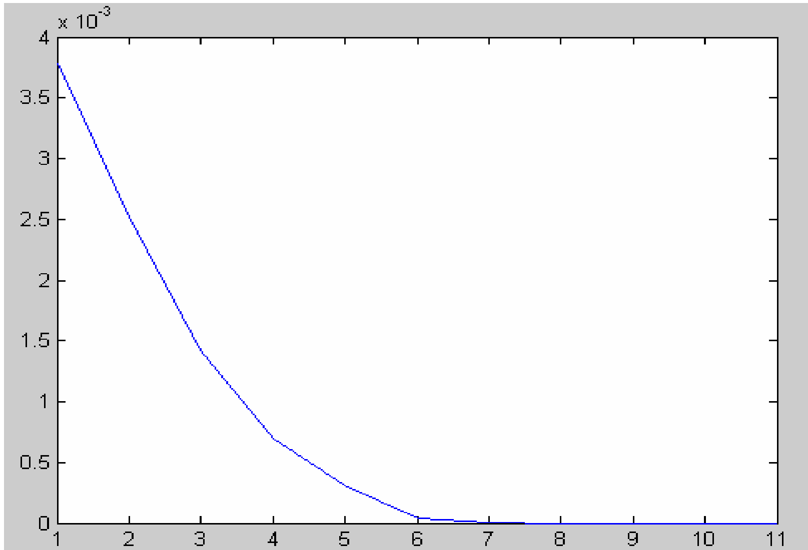
rxmccdmarake4.m

```
function bitsalah=rxmccdmarake4(data,yreceive,N,jumsubcar,kode,rake,jum,
                                        f,dt,t,pjkode)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%siyal terima
ypath=SerToPar(yreceive,pjkode*N*jum);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ydmod=zeros(jumsubcar,length(ypath));
ydepulse=zeros(rake,length(data)*pjkode,jumsubcar);
yfft=zeros(rake,length(data)*pjkode,jumsubcar);
ydespread=zeros(rake,length(data)*pjkode,jumsubcar);
yrake=zeros(rake,length(data),jumsubcar);

% proses demodulasi sinyal
for i=1:jumsubcar
   ydmod(i,:)=ypath(i,:).*cos(2*pi/jum*f(i).*t);

   % proses depulse
   for k=1:rake
      ydepulse(k,:,i)=intdump(ydmod(i,:),jum);

   % proses fft
     for h=1:N
```

```
      yfft(k,pjkode*h-(pjkode-1):pjkode*h,i)=fft(ydepulse(k,pjkode*h-(pjkode
                                              -1):pjkode*h,i),pjkode);
    end

  % proses despreading dengan metode EGC tiap rake
    bn=1; % EGC
    delaykode(k,:,i)=bn*[zeros(1,dt(k)) kode(i,1:length(kode)-dt(k))];
    ydespread(k,:,i)=despreading(yfft(k,:,i),pjkode,delaykode(k,:,i));

  % proses integral
     for h=1:length(data)
        yrake(k,h,i)=integral(ydespread(k,pjkode*h-(pjkode-1):pjkode*h,i));
     end

  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Proses decision tiap jumsubcar
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if rake==1
  for i=1:jumsubcar
     ysubcar(i,:)=yrake(:,:,i);
  end
else
  for i=1:jumsubcar
     ysubcar(i,:)=sum(yrake(:,:,i));
  end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   keluaran akhir masing subcarrier
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yakhir=limiter(sum(ysubcar));
```

```
%data
%keluaran dalam jumlah bit yang salah
bitsalah=symerr(yakhir,data);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%sinyal noise tampa multipath fading
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%sinyal user satu
yinti=Rx(1,:);                    % interference user yg lain
yinterfrc=sum(Rx(2:end,:),1);     %penambahan noise
yrx=yinti+0.1.*yinterfrc;
axix=1;
   for SNR=1:5:40
      yreceive=awgn(yrx,SNR,0);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% proses dipenerima user 1 sebagai acuan.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%prose pemanggilan function rxmccdmarake4
jumbiterr(loop,axix)=rxmccdmarake4(data(1,:),yreceive,N,jumsubcar,kode(:,:,1),
                                  rake,jum,f,dt,t,pjkode);
% keluaran dalam bentuk jumlah bit yang error
      axix=axix+1;
   end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% proses dipenerima user 1 sebagai acuan.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
jumbiterr(loop,axix)=rxmccdmarake4(data(1,:),yreceive,N,jumsubcar,kode(:,:,1),
                                  rake,jum,f,dt,t,pjkode);
%% keluaran dalam bentuk jumlah bit yang error
%       axix=axix+1;
%    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

```
toc
%jumlah error bit yg terjadi terhadap total data yang dikirim
Bittotsalah=sum(jumbiterr,1);
BERR=Bittotsalah/N/loop;
multiuserrake4=BERR;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Hasil Plot BER.
if rake==1
   semilogy(BERR,'k-'),grid on,hold on
elseif rake==3
   semilogy(BERR,'b-'),grid on,hold on
elseif rake==4
   semilogy(BERR,'b-'),grid on,hold on
elseif rake==5
   semilogy(BERR,'r-'),grid on,hold on
else rake==6
   semilogy(BERR,'k-'),grid on,hold on
end
legend('1 user')
title('Grafik Perbandingan SNR dengan BER');
xlabel('SNR');
ylabel('BER');
```



Gambar D-5 Hasil Simulasi Multiuserrake4