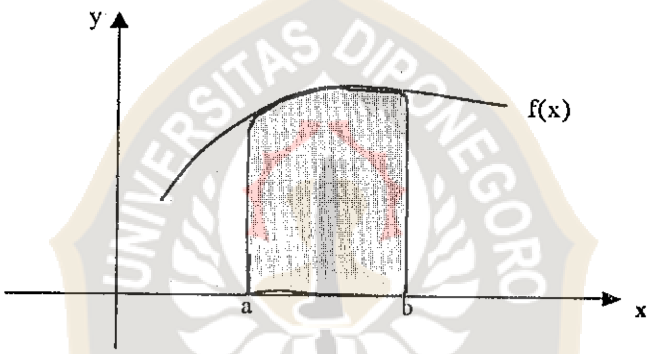
BAB II

TEORI PENUNJANG

2.1. Integral Tertentu

Diberikan sebuah daerah A terletak pada interval [a,b] pada sumbu x yang dibatasi oleh sebuah fungsi kontinyu f(x) dan sumbu x.



Gambar 2.1. Daerah A dengan pembatas f(x) pada [a,b]

Sebuah partisi P pada interval tertutup [a,b] akan membuat suatu himpunan bagian berhingga dari interval [a,b] yang memuat titik a dan b.

$$P = \{x_0, x_1, \ldots, x_n\} \tag{2.1}$$

P merupakan partisi dari [a,b] dengan $x_0 = a$ dan $x_n = b$, yang tersusun atas

$$a = x_0 < x_1 < \ldots < x_n = b \tag{2.2}$$

Jika $P = \{x_0, x_1, \ldots, x_n\}$ adalah partisi dari [a,b] maka P memecah interval [a,b] menjadi sejumlah sub-interval.

$$[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n] \tag{2.3}$$

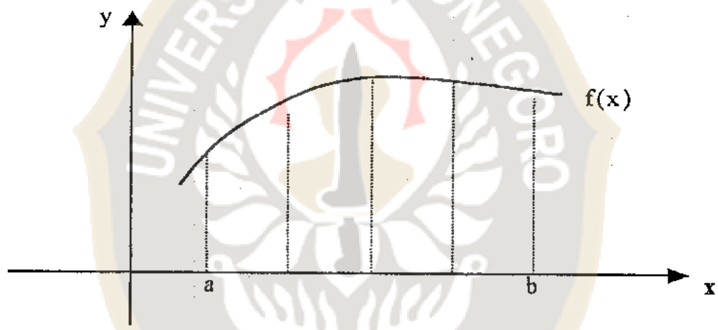
dengan lebar subinterval $\Delta x_i = x_i - x_{i-1}$ untuk $i = 1,2,\ldots,n$.

Anggap bahwa f merupakan fungsi kontinyu pada interval [a,b], maka pada setiap subinterval $[x_i,x_{i-1}]$ fungsi f memuat nilai minimum dan nilai maksimum (m_i dan M_i)

$$m_i = \inf\ (\ f(x) : x \in [x_i,x_{i-1}]\ )\ \text{dan} \tag{2.4}$$

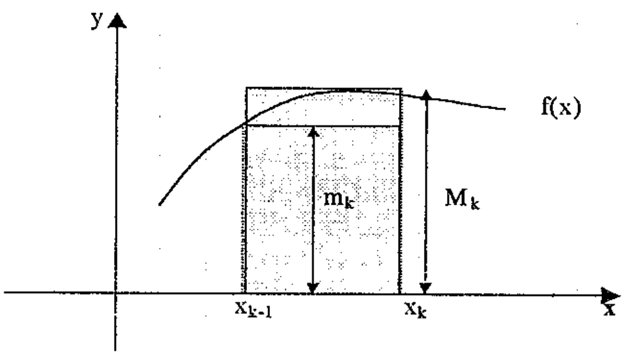
$$M_i = \sup\ (\ f(x) : x \in [x_i,x_{i-1}]\ )\ \ \text{untuk } i = 1,2,\ldots,n \tag{2.5}$$

Partisi P akan membuat daerah A terbagi menjadi sub-daerah A_i dengan $i = 1, 2, \ldots, n$



Gambar 2.2. Subdaerah- subdaerah A

Ambil suatu subdaerah A_k, $1 \leq k \leq n$ pada interval $[x_{k-1},x_k]$



Gambar 2.3. Sub-daerah A_{k1} dan A_{k2}

Dengan menggunakan (2.4) dan (2.5), luas daerah A_{k1} dan A_{k2} dapat dihitung, yaitu :

$$\text{Luas } A_{k1} = m_k * (x_k - x_{k-1}) \quad (2.6)$$

$$\text{Luas } A_{k2} = M_k * (x_k - x_{k-1}) \quad (2.7)$$

Sementara itu berdasarkan gambar (2.3), luas subdaerah A_k dapat dinyatakan dalam pertidaksamaan :

$$\text{Luas } A_{k1} \leq \text{Luas } A_k \leq \text{Luas } A_{k2} \quad (2.8)$$

$$m_k (x_k - x_{k-1}) \leq \text{Luas } A_k \leq M_k (x_k - x_{k-1}) \quad (2.9)$$

Jika setiap sub-daerah dilakukan hal yang sama maka diperoleh :

$$L_f(P) = m_1 \Delta x_1 + m_2 \Delta x_2 + \ldots + m_n \Delta x_n \quad (2.10)$$

yang disebut senbagai jumlah bawah fungsi f dengan partisi P (lower sum), dan

$$U_f(P) = M_1 \Delta x_1 + M_2 \Delta x_2 + \ldots + M_n \Delta x_n \quad (2.11)$$

disebut jumlah atas fungsi f dengan partisi P (upper sum)

dengan $\Delta x_i = x_i - x_{i-1}$ untuk i= 1,2,...,n

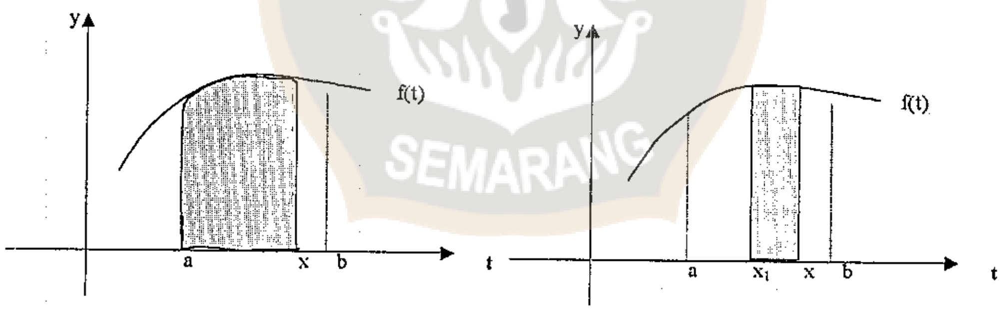
Definisi 2.1

Bilangan unik I yang memenuhi pertidaksamaan $L_f(P) \leq I \leq U_f(P)$ disebut integral tertentu dari fungsi f pada interval [a,b] dan dinyatakan oleh $I = \int_a^b f(x)dx$.

2.1.1. Luas Dalam Integral

Ilustrasi untuk menggambarkan luas daerah A, suatu daerah yang terletak dibawah kurva f(x) dan sumbu x dalam interval [a,b], diberikan suatu daerah yang disapu oleh garis vertikal pada sumbu x dimana $a \le x \le b$, bergerak dari kiri ke kanan mulai titik x = a sampai x = b. Garis vertikal ini membujur dari sumbu x sampai kurva y = f(x), sehingga panjang garis ini tentunya sama dengan f(x).

Daerah sapuan tergantung pada nilai x yang diberikan, sehingga merupakan suatu fungsi dari x, misal A(x). A(x) berarti luas daerah sapuan A dari a sampai x.



(a) Daerah sapuan A(x) (b) Daerah antara x_1 dan x

Gambar 2.4. Daerah Sapuan A(x) dengan fungsi f(x)

Teorema 2.1

Jika f kontinyu pada [a,b], Fungsi A terdefinisi pada interval [a,b] dinyatakan oleh $A(x) = \int_a^x f(t)dt$ adalah kontinyu pada [a,b], maka A(x) dapat diturunkan pada interval (a,b) dan turunannya $A'(x) = f(x)$ untuk $\forall$ x dalam (a,b).

Bukti :

Kita mulai dengan suatu pengertian tentang turunan suatu fungsi F pada suatu titik x_1 :

$$F'(x_1) = \lim_{x \to x_1} \frac{F(x) - F(x_1)}{x - x_1} \tag{2.12}$$

Misal x_1 adalah sembarang bilangan pada interval (a,b). Untuk sembarang bilangan $x > x_1$ maka $A(x) > A(x_1)$ adalah daerah dibawah kurva f(x) antara x_1 dan x. Jika $f(x_m)$ merupakan nilai minimum f pada interval $[x_1,x]$, dan $f(x_M)$ adalah nilai maksimum pada interval $[x_1,x]$, maka berdasarkan (2.6), (2.7) terdapat pertidaksamaan

$$f(x_m)\ (x - x_1) \le A(x) - A(x_1) \le f(x_M)\ (x - x_1) \tag{2.13}$$

karena $x - x_1 > 0$ maka

$$f(x_m) \le \frac{A(x) - A(x_1)}{x - x_1} \le f(x_M) \tag{2.14}$$

fungsi f kontinyu dan $x_1 \le x_m \le x$, $x_1 \le x_M \le x$, sehingga

$$\lim_{x \to x_1} f(x_m) = f(x_1) \quad \text{dan} \quad \lim_{x \to x_1} f(x_M) = f(x_1) \tag{2.15}$$

Oleh karena limit keduanya pada (2.15) adalah sama, maka

$$\lim_{x \to x_1} \frac{A(x) - A(x_1)}{x - x_1} = f(x_1) \tag{2.16}$$

Jadi berdasarkan definisi (2.12) dan (2.16), karena x_1 adalah sembarang nilai pada (a,b) maka

$A'(x) = f(x)$. (terbukti).

Definisi 2.2. Antiderifativ

Suatu fungsi G disebut antiderifativ untuk fungsi f jika dan hanya jika G kontinyu pada [a,b] dan $G'(x) = f(x)$ untuk $\forall x \in (a,b)$

Teorema 2.2

Misal f kontinyu pada [a,b]. Jika G antiderifativ untuk f pada [a,b]

maka $$\int_a^b f(t)dt = G(b) - G(a)$$

Bukti :

Menurut Teorema (2.1) diketahui bahwa

$$A(x) = \int_a^x f(t)dt$$

A(x) adalah antiderifativ dari f pada interval [a,b]. Jika G juga merupakan antiderivatif untuk f pada interval [a,b], maka A dan G kontinyu pada [a,b] dan memenuhi $A'(x) = G'(x)$ untuk $\forall$ x pada (a,b).

Jika G adalah sembarang antiderifatif dari f, dan terdapat konstanta C sedemikian hingga

$$A(x) = G(x) + C \tag{2.17}$$

Karena A(x) untuk x = a, A(a) = 0, maka

$$A(a) = G(a) + C$$

$$0 = G(a) + C$$

$$C = -G(a) \tag{2.18}$$

Jadi $$A(x) = G(x) - G(a) \tag{2.19}$$

Untuk x = b, diperoleh :

$$A(b) = G(b) - G(a) \quad \underline{\text{(terbukti)}}$$

Sifat-sifat integral berhingga :

1. Jika $f(x) \geq 0$ untuk $\forall\, x \in [a,b]$ maka $\int_a^b f(x)dx \geq 0$ (2.20)

2. Jika $f(x) \geq g(x)$ untuk $\forall\, x \in [a,b]$, maka $\int_a^b f(x)dx \geq \int_a^b g(x)dx$ (2.21)

Contoh 2.1.1

Misal diberikan integral $I = \int_1^3 x^3 dx$

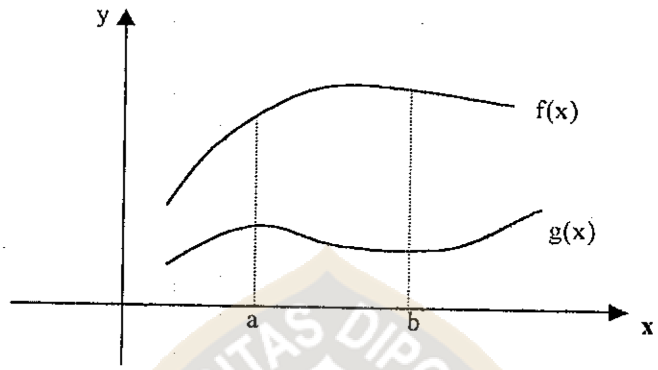
Dari soal diatas diketahui bahwa $f(x) = x^3$. sehingga menurut teorema 2.2 antiderivatifnya adalah

$$G(x) = \tfrac{1}{4} x^4$$

sehingga solusinya :

$$\begin{aligned} I = A(x) &= G(3) - G(1) \\ &= \tfrac{1}{4}(3)^4 - \tfrac{1}{4}(1)^4 \\ &= 20 \text{ satuan} \end{aligned}$$

2.1.2. Luas Daerah Diantara Dua Kurva



Gambar 2.5. Daerah diantara f(x) dan g(x)

Teorema 2.3.

Jika $f(x) \geq g(x) > 0$ dan kedua fungsi adalah kontinyu untuk $\forall$ x pada interval [a,b], Luas daerah antara $y_1 = f(x)$ dan $y_2 = g(x)$ adalah

$$I = \int_a^b [f(x) - g(x)]dx$$

<u>Bukti</u> :

Dari Gambar 2.5 jelas terlihat bahwa I_1 adalah luas daerah untuk $y = f(x)$ pada interval [a,b]. Daerah A_2 adalah daerah yang dibatasi kurva g(x) pada interval [a,b]. Berdasarkan sifat-sifat integral berhingga (2.21) dan dari gambar terlihat bahwa $I_1 \geq I_2$, sehingga

$$I = I_1 - I_2 \tag{2.22}$$

$$I = \int_a^b f(x)dx - \int_a^b g(x)dx \tag{2.23}$$

$$I = \int_a^b [f(x) - g(x)]dx \qquad \text{(Terbukti)}$$

Teorema 2.4.

Jika f(x) ≥ g(x) dan g(x) < 0 untuk ∀ x pada [a,b], kedua fungsi kontinyu pada interval [a,b], maka luas daerrah diantara f(x) dan g(x) pada interval [a,b] adalah :

$$I = \int_a^b [f(x) - g(x)]dx$$

Bukti :

Ambil konstanta k, sedemikian hingga g(x) + k > 0. Tambahkan juga k pada f(x) sehingga terbentuk daerah baru yang kongruen dengan daerah semula, tetapi berada diatas sumbu x untuk ∀ x pada interval [a,b].

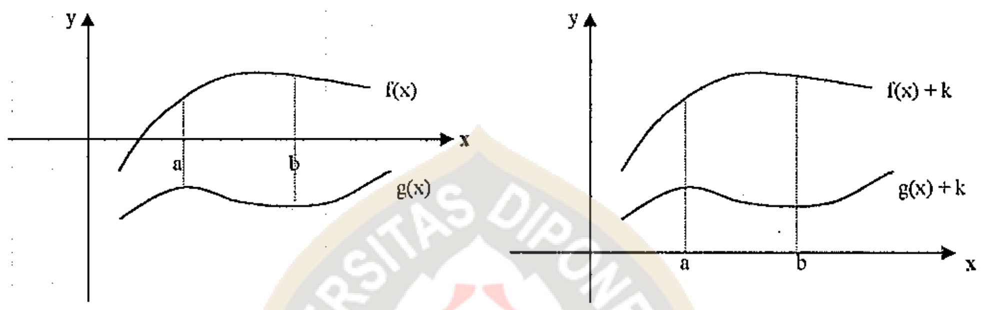
Sehingga diperoleh fungsi fungsi baru :

$$f(x) + k \geq g(x) + k > 0 \tag{2.24}$$

jadi, berdasarkan (2.24) luas daerah baru adalah

$$I = \int_a^b [(f(x)+k)-(g(x)+k)]dx \qquad (2.25)$$

$$I = \int_a^b [f(x)-g(x)]dx \qquad \underline{\text{(Terbukti)}}.$$



Gambar 2.6 (a) $f(x) \geq g(x) < 0$ Gambar 2.6 (b) $f(x)+k \geq g(x)+k > 0$

Contoh 2.1.2

Hitung luas daerah diantara dua kurva $y_1 = 5 - x^2$ dan $y_2 = 3 - x$

Jawab :

y_1 dan y_2 berpotongan dititik $x = -1$ dan $x = 2$.

untuk $\forall x \in [-1,2] \rightarrow y_1 \geq y_2 > 0$

sehingga luas daerah yang dibatasi oleh kurva y_1 dan y_2 adalah :

$$\int_{-1}^{2} (5 - x^2) - (3 - x)dx$$

$$\int_{-1}^{2} (2 + x - x^2)dx = \left[2x + \frac{1}{2}x^2 - \frac{1}{3}x^3\right]_{-1}^{2}$$

$$= 5/2 \text{ satuan}$$

2.2. Himpunan

Himpunan merupakan kumpulan dari obyek-obyek yang dapat dinyatakan secara jelas. Obyek-obyek yang terdapat dalam himpunan disebut anggota atau elemen. Simbol $\in$ menandakan elemen dari himpunan.

Contoh 2.2.1. P = {himpunan dari semua bilangan prima}

$2 \in P$, $3{,}5 \in P$, $6 \notin P \rightarrow$ bilangan enam bukan bilangan prima.

Definisi 2.3. Himpunan Bagian

Himpunan A merupakan himpunan bagian dari B jika setiap elemen dari A juga merupakan elemen dari B.

Himpunan bagian dinotasikan dengan simbol $\subseteq$. Sehingga $A \subseteq B$ berarti A himpunan bagian dari B.

Contoh 2.2.2. Z = {himpunan semua bilangan bulat}

N = {himpunan semua bilangan kelipatan dua}

Karena semua bilangan kelipatan 2 adalah bilangan bulat maka $N \subseteq Z$.

Definisi 2.4. Himpunan Semesta atau Semesta Pembicaraan

Himpunan semesta adalah himpunan dari semua elemen-elemen yang sedang dibicarakan.

Contoh 2.2.3 U = {himpunan dari semua bilangan riil}

V = {himpunan dari semua titik-titik dalam bidang datar}

U dan V adalah himpunan universal (*universal set*)

Definisi 2.5. Komplemen

Komplemen dari sembarang himpunan A adalah himpunan dari elemen-elemen terletak pada himpunan universan dan bukan anggota A.

Komplemen dari himpunan A dinotasikan dengan A'

$A' = \{x \mid x \in U \text{ dan } x \notin A\}$

Karena x selalu anggota dari U maka komplemen A sering dinyatakan dengan

$A' = \{x \mid x \notin A\}$

Contoh 2.2.4.

U = {himpunan dari semua huruf-huruf alpabetik}

A = {himpunan dari semua huruf mati}

C = {himpunan dari semua huruf hidup}

Jadi C = A'

Definisi 2.6. Gabungan

Gabungan dari dua himpunan A dan B adalah himpunan dari semua elemen yang terletak pada A atau terletak pada B.

$A \cup B = \{x \mid x \in A \text{ atau } x \in B\}$

Contoh 2.2.5

A = {1,2,3,4,5,}   B = {3,4,5,6,7,8}

$A \cup B$ = {1,2,3,4,5,6,7,8}

Definisi 2.7. Irisan

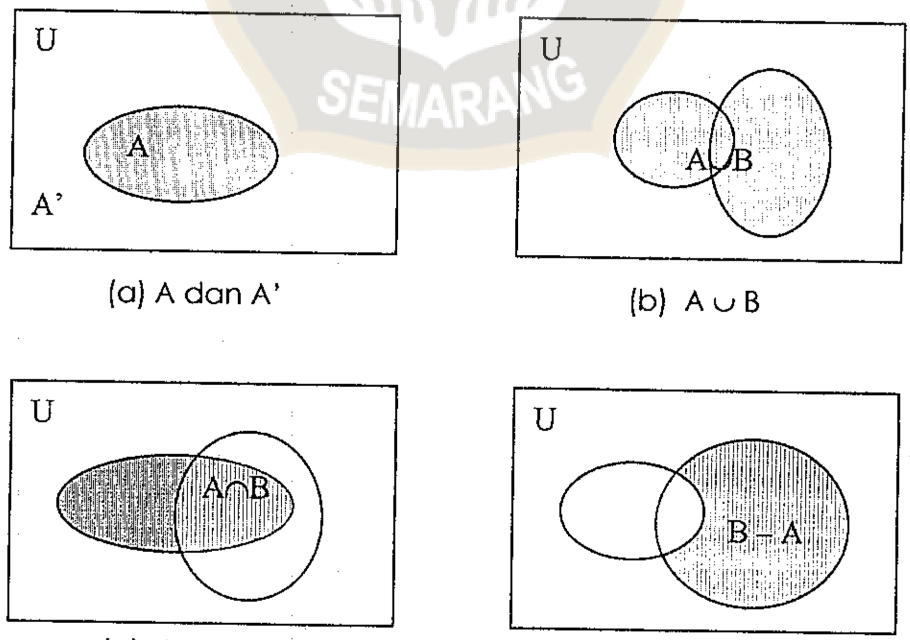
Irisan dari dua himpunan A dan B adalah himpunan dari semua elemen yang terletak pada A dan juga terletak pada B.

$A \cap B = \{ x \mid x \in A \text{ dan } x \in B \}$

Contoh 2.2.6

Seperti pada contoh 2.2.5 $A \cap B$ = {3,4,5}

Ilustrasi dengan menggunakan diagram venn untuk operasi himpunan ditunjukkan oleh Gambar 2.7 berikut ini.



(a) A dan A'   (b) $A \cup B$

(c) $A \cap B$ (d) B – A

Gambar 2.7. Operasi-operasi himpunan

Pada Gambar 2.7 (d) terdapat himpunan hasil operasi pengurangan B – A yang didefinisikan oleh

$$B - A = \{ x \mid x \in B \text{ dan } x \notin A \} \quad (2.26)$$

Atau dengan menggunakan definisi irisan :

$$B - A = (B \cap A') \quad (2.27)$$

Definisi 2.8

Suatu ring dari himpunan-himpunan adalah kumpulan tak kosong R dari himpunan-himpunan yang memenuhi :

a. jika $S, T \in R$ maka $S \cup T \in R$

b. jika $S,T \in R$ maka $S - T \in R$

Ring dari himpunan mempunyai sifat tertutup terhadap gabungan dan selisih. Akibatnya ring tersebut juga tertutup terhadap irisan.

$$S \cap T = S - (S - T) \quad (2.28)$$

Selanjutnya R diasumsikan sebagai kumpulan dari himpunan bagian himpunan bagian dari himpunan U.

Definisi 2.9

Suatu ukuran di dalam U adalah suatu fungsi bernilai nyata m didefinisikan pada ring R dari himpunan bagian pada V dan memenuhi :

a. $\forall S \in R,\ m(S) \geq 0$

b. $\forall S, T \in R$, jika $S \cap T = \varnothing$ maka $m(S \cup T) = m(S) + m(T)$

Ukuran suatu himpunan dimaksudkan sebagai banyaknya elemen atau anggota himpunan tersebut.

Contoh 2.2.7

$V = Z$: {semua bilangan bulat}

R adalah kumpulan dari himpunan bagian yang berhingga dari V untuk $S \in R$, maka $m(S)$ adalah jumlah anggota dari S.

Teorema 2.5

Jika $S,T \in R$ dan $S \subseteq T$ maka $m(T - S) = m(T) - m(S)$

Bukti.

Dari definisi Pengurangan diketahui bahwa

$x \in (T - S) = \{x \in T \text{ dan } x \notin S\}$ atau

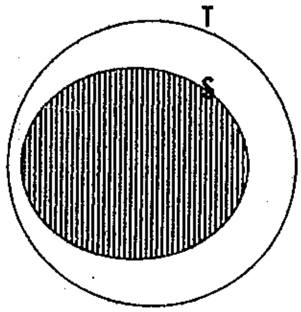
$$T - S = T \cap S' \text{ (lihat gambar)} \tag{2.29}$$

berdasarkan Gambar 2.8 dan (2.29)

$$S \cap (T - S) = S \cap (T \cap S') = \varnothing \tag{2.30}$$

$$S \cup (T - S) = S \cup (T \cap S') \tag{2.31}$$

karena $S \subseteq T$ dan menurut definisi sub set maka $S \cup (T \cap S') = T$



Gambar 2.8. $S \subseteq T$

$$\text{sehingga} \quad m(T) = m(S) \cup m(T \cap S') \tag{2.32}$$

$$= m(S) \cup m(T - S) \tag{2.33}$$

$$= m(S) + m(T - S) \tag{2.34}$$

jadi diperoleh m(T) – m(S) = m(T-S). Terbukti

Contoh 2.2.8

S = {himpunan bilangan prima antara 1 sampai 10}

T = {himpunan bilangan bulat antara 1 sampai 10}

dengan demikian tampak bahwa $S \subseteq T$

S = {2,3,5,7} m(S) = 4

T = {1,2,3,4,5,6,7,8,9,10} m(T) = 10

maka himpunan bilangan bulat bukan prima antara 1 sampai 10 adalah :

T – S = {1,4,6,8,9,10}

$$m(T - S) = m(T) - m(S)$$

$$= 10 - 4 = 6.$$

2.3. Distribusi Uniform

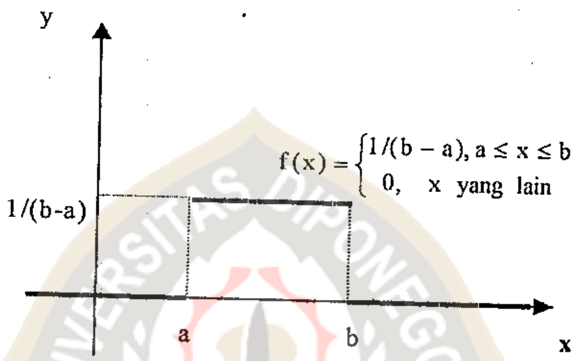
Distribusi Uniform merupakan distribusi yang paling sederhana untuk variabel random kontinyu.

Definisi 2.10

Jika fungsi kepadatan peluang dari variabel random x diberikan oleh

$$f(x) = \begin{cases} 0 & x < a, x > b \\ \frac{1}{b-a} & a \le x \le b \end{cases} \tag{2.35}$$

dimana parameter a dan b terletak pada $-\infty < a < b < \infty$, maka variabel random x yang didefinisikan untuk berdistribusi uniform dalam interval [a,b] dan distribusinya diberikan oleh (2.35) disebut Distribusi Uniform.



Gambar 2.5. Fungsi kepadatan peluang distribusi uniform

2.4. Bilangan Random

Bilangan random adalah kejadian khusus dari variabel random. Pada umumnya bilangan random banyak digunakan dalam perhitungan Monte Carlo serta proses random yang membutuhkan bilangan random dengan suatu distribusi tertentu. Menurut cara pembangkitannya terdapat 2 (dua) perbedaan bilangan random, yaitu bilangan random asli (dibangkitkan secara alami) dan buatan (dibangkitkan dengan campur tangan manusia). Pada bilangan random yang kedua inilah yang banyak digunakan karena dapat dimodifikasi sesuai keperluan sistem.

1. **Bilangan Random Sebenarnya (truly-random)**

Bilangan random sebenarnya (truly) hanya dapat dibangkitkan oleh proses fisis, misalnya peluruhan radioaktif , roulet dan lain-lain. Bilangan random jenis ini sepenuhnya tidak dapat diprediksi bilangan berapa yang akan muncul berikutnya sehingga mempersulit untuk dipakai dalam proses perhitungan, karena harus dibangkitkan oleh peralatan terpisah baru kemudian dicatat atau direkam dengan pita magnetik. Salah satu contoh pita magnetik yang berisi bilangan random hasil peluruhan diproduksi oleh Argone National Laboratory Code Center, Argone, Illinois, Amerika Serikat. Pita ini memuat 2,5 juta bilangan random ukuran 32 bit.

2. **Bilangan Pseudorandom (Buatan)**

Bilangan Pseudorandom merupakan bilangan yang paling sering digunakan dalam perhitungan Monte Carlo. Bilangan ini dibangkitkan oleh suatu algoritma numeris yang sering dikenal dengan Linier Congruential Generator. Oleh karena itu bilangan ini sepenuhnya dapat diprediksi berapa yang akan muncul berikutnya, jika algoritma beserta konstanta yang digunakan, seperti pada persamaan (2.36), untuk membangkitkannya diketahui. Tetapi bilangan ini dapat dianggap bilangan random sebenarnya bagi orang yang tidak mengetahui algoritma (LCG) yang digunakan.

Untuk lebih lengkap mengenai cara-cara pembangkitan bilangan random akan dijelaskan dalam sub-bab berikut ini.

2.4.1. Pembangkitan Bilangan random

Bilangan random dapat dibangkitkan dengan cara manual (roulet, dadu, peluruhan radioaktif) untuk bilangan Truly-random atau dengan suatu algoritma numeris untuk bilangan Pseudorandom. Pada umumnya prosedur pembangkitan bilangan pseudorandom menggunakan hubungan kesebangunan (congruence) dan relasi rekursif. Prosedur pembangkitan atau generator yang dipakai pada umumnya menggunakan nilai awal (seed) sebagai bilangan awal untuk mencari bilangan kedua. Kemudian bilangan kedua ini dimasukan kedalam prosedur lagi untuk mencari bilangan ketiga, demikian seterusnya.

Prosedur yang banyak dipakai dalam pembangkitan bilangan pseudorandom adalah *Linier Congruential Generator* yang disusun oleh **Lehmer (R.K. Bock, 1998).** Secara matematis prosedur ini dinyatakan dengan :

$$R_{i+1} = (p\, R_i + c) \text{ modulo } m \qquad (2.36)$$

Konstanta p, c dan m merupakan suatu tetapan yang harus ditentukan terlebih dahulu. Persamaan (2.36) diatas menyatakan untuk mendapatkan bilangan ke-i adalah dengan cara mengalikan bilangan ke i-1 dengan konstanta p ditambahkan dengan c dan mendapatkan sisanya setelah menbaginya dengan m. Untuk memulai perhitungan ditentukan suatu bilangan awal sebagai nilai awal (seed) yaitu R_0.

Dengan demikian spesifikasi lengkap prosedur pembangkit bilangan Pseudorandom diatas membutuhksn konstanta R_0, p, c dan m.

Persamaan (2.36) dapat dimodifikasi menjadi beberapa prosedur tergantung pada pemilihan konstanta p dan c.

- Jika diambil p = 1 maka prosedur (2.36) menjadi prosedur penjumlahan,
- jika dipilih c = 0 dan p, maka dinamakan prosedur perkalian (multiplicative congruential generator).

1. Metode Penjumlahan

Untuk $p = 1$, $c \neq 0$ diperoleh

$$R_{i+1} = (R_i + c) \text{ Mod } m$$

Contoh 2.4.1

$R_0 = 2$, $c = 4$, $m = 15$

bilangan random yang muncul ;

2, 6, 10, 14, 3, 7, 11, 0, . . .

Pada metode ini bilangan yang muncul berikutnya cenderung mudah untuk ditebak dengan mengamati pola barisan yang terbentuk.

2. Metode Perkalian

Metode ini mengambil c = 0, sehingga

$$R_{i+1} = (p * R_i) \text{ Mod } m \qquad (2.37)$$

Pada umumnya modulo m dipilih 2^k atau 2^k-1 dengan $k \geq 2$ karena bilangan yang disusun komputer berada dalam sisitem biner (0 dan 1), sehingga untuk memaksimalkan kapasitas semua k-bit harus dimanfaatkan. Konstanta p dan nilai awal R_0 sebaiknya ganjil, karena jika salah satu genap (2n) maka sesuai dengan sifat perkalian 2 (dua) bilangan asli, bilangan yang dihasilkan juga genap yang mempunyai kemungkinan memperpendek daur perulangan barisan.

Menurut Hamming sebaiknya $p = 8t \pm 3$, untuk suatu t.

Contoh 2.4.2. untuk k = 5 (menggunakan 5 bit sistem biner)

misal p $= 10101_2 = 21$,

$R_0 = 10001 = 17$

maka diperoleh $R_{i+1} = (21 * R_i) \bmod 32$

Tabel 1. Bilangan pseudo random

Integer	Biner
$R_0 = 17$	10001
$R_1 = 5$	00101
$R_2 = 9$	01001
$R_3 = 29$	11101
$R_4 = 1$	00001
$R_5 = 21$	10101
$R_6 = 25$	11001
$R_7 = 13$	01101
$R_8 = 17$	10001

Dalam sistem bilangan dikenal adanya notasi :

$$x \equiv p \ (\mathrm{mod}\ m) \qquad (2.38)$$

yang berarti x – p dapat dibagi m.

Jika tersedia k-digit biner dan berdasar (2.36) dan (2.37) diperoleh

$$R_{i+1} \equiv p^{*}R_0 \ \ (\mathrm{Mod}\ \ 2^k) \qquad (2.39)$$

Konstanta p sebaiknya ganjil, karena jika genap (2m) barisan yang muncul :

$$R_1 = p * R_0$$

$$R_2 = p * R_1 = p^2 * R_0$$

$$R_3 = p * R_2 = p^3 * R_0$$

$$R_n = p * R_{n-1} \quad = p^n * R_0 \quad = (2m)^n * R_0$$

$$= (2^n\ m^n)\ R_0$$

dimana terdapat faktor 2^n sehingga pada saat n = k, R_n = 0 dan untuk selanjutnya bilangan yang muncul adalah 0.

Untuk suatu bilangan ganjil dapat ditulis dalam salah satu bentuk berikut ini :

8t – 3 8t – 1 8t + 1 8t + 3

untuk suatu t.

Tabel 1 menunjukkan percobaan yang dilakukan untuk membangkitan bilangan pseudorandom dengan k-digit biner (k=5) dihasilkan sebanyak 2^{k-2} suku-suku sebelum terjadi perulangan dengan suku awal. Jika 2 (dua) digit terakhir dihilangkan (karena 2 digit terakhir semuanya sama yaitu '01'), maka dapat dihasilkan barisan bilangan pseudorandom dengan permutasi dari 0, 1, 2, . . .,

2^{k-2}-1 secara penuh dengan panjang 2^{k-2}. Hal ini disebabkan karena 2 (dua) digit terakhir tersebut selalu '01' sehingga tidak dapat dikatakan random.

3. Metode Campuran (Mixed Congruential Generator)

Prosedur pembangkit bilangan pseudorandom sebaiknya mempunyai karakteristik sebaga berikut :

- Routine (perulangan) harus cepat
- Efisien dalam penyimpanan (tidak memerlukan memori yang besar)
- Barisan bilangan yang dihasilkan mempunyai daur yang panjang.

Dengan menggunakan persamaan (2.36) akan dilakukan percobaan untuk mendapatkan 16 bilangan random.

Contoh 2.4.3.

Misal konstanta yang dipilih p = 4, c = 5 dan m = 16, sehingga persamaannya menjadi

$$R_{i+1} = (4 * R_i + 5) \bmod 16 \tag{2.40}$$

Nilai awal (R_0) = 3

$$R_1 = (3 * 4 + 5) \bmod 16 = 1$$

$$R_2 = (1 * 4 + 5) \bmod 16 = 9$$

$$R_3 = (9 * 4 + 5) \bmod 16 = 9$$

$$R_4 = (9 * 4 + 5) \bmod 16 = 9$$

Demikian seterusnya , bilangan yang diperoleh adalah 9. Jadi barisan bilangan randomnya 1, 9, 9, ..., 9,... .

Contoh 2.4.4

Diambil $p = 5$, $c = 6$ dan $m = 8$

Prosedur pembangkit bilangan randomnya

$$R_{i+1} = (5^*R_i + 6) \bmod 8 \tag{2.41}$$

Diberikan nilai awal $R_0 = 4$.

$$R_1 = (4 * 5 + 6) \bmod 8 = 3$$

$$R_2 = (3 * 5 + 6) \bmod 8 = 5$$

$$R_3 = (5 * 5 + 6) \bmod 8 = 7$$

$$R_4 = (7 * 5 + 6) \bmod 8 = 1$$

$$R_5 = (1 * 5 + 6) \bmod 8 = 3$$

$$R_6 = (3 * 5 + 6) \bmod 8 = 5$$

Bilangan yang muncul berikutnya adalah 7, 1, 3,...,dst. Dari kedua percobaan diatas diketahui bahwa barisan bilangan random mengalami pengulangan. Pada contoh 1, hanya ada 2 (dua) macam bilangan yang muncul yaitu 1 dan 9. Sedangkan pada contoh 2, bilanagn random yang dihasilkan adalah 1, 3, 5 dan 7. Selebihnya bilangan tersebut akan muncul berulang-ulang.

2.4.2. Pembangkitan Bilangan Pseudorandom Berdistribusi Uniform

Distribusi uniform mempunyai pengertian secara umum bahwa probabilitas suatu variabel x yang berada pada suatu interval tertentu sebanding dengan rasio antara ukuran sub-interval (jarak antar titik) dan jangkauannya (lebar interval). Dengan kata lain setiap titik dalam jangkauan tersebut mempunyai kemungkinan yang sama untuk dipilih. Misalkan jangkauan nilai-nilai yang mungkin adalah dari A ke B (B > A), maka probabilitas x berada dalam interval Δx adalah Δx / (B-A).

Berdasarkan contoh-contoh pada sub-bab sebelumnya serta Lampiran 1, dapat dikatakan bahwa pemilihan konstanta p, c, m dan R_0 tidak dapat dilakukan secara sembarangan. Seleksi yang teliti pada setiap konstanta akan menghasilkan suatu barisan bilangan random dengan daur yang panjang. Persoalan memilih konstanta–konstanta merupakan persoalan rumit dan tidak ada cara-cara analitik untuk memperoleh tetapan-tetapan yang bagus. Beberapa tetapan yang telah ditemukan merupakan hasil percobaan yang berulang-ulang.

Menurut Fuller (1998), yang melakukan percobaan untuk mendapatkan 16 (enam belas) bilangan random, ada beberapa syarat yang harus dipenuhi, yaitu :

- modulus m > 0
- faktor perkalian (p) $0 \le p \le m$
- faktor pertambahan (c) , $0 \le c \le m$

- nilai awal $R_0 \geq 0$

kemudian syarat-syarat khususnya adalah :

1. faktor pertambahan (c) bukan faktor dari 16.
2. p = b + 1, dengan b adalah kelipatan dari k dimana k suatu bilangan prima faktor dari 16. Konstanka k diperoleh = 2, sehingga b = 2, 4, 6, 8, 10, 12, 14.
3. jika m kelipatan 4 maka b juga kelipatan dari 4. Karena 16 merupakan kelipatan 4 maka b = 4, 8, 12.

Dengan menggunakan ketiga syarat diatas diambil c = 3, b = 4 → p = 5 sehingga prosedur pembangkitnya menjadi

$$R_{i+1} = (5*R_i + 3) \bmod 16 \tag{2.42}$$

Persamaan (2.42) akan membangkitkan penuh bilangan antara 0 sampai 15.

Nilai awal R_0 bebas dipilih dengan syarat $0 \leq R_0 < 16$. Misalkan dipilih R_0 = 3.

$R_1 = (3*5 + 3) \bmod 16 = 2$

$R_2 = (2*5 + 3) \bmod 16 = 13$

$R_3 = (13*5 + 3) \bmod 16 = 4$

$R_4 = (4*5 + 3) \bmod 16 = 7$

$R_5 = (7*5 + 3) \bmod 16 = 6$

$R_6 = (6*5 + 3) \bmod 16 = 1$

$R_7 = (1*5 + 3) \bmod 16 = 8$

$R_8 = (8*5 + 3) \bmod 16 = 11$

$R_9 = (11*5 + 3) \bmod 16 = 10$

$R_{10} = (10*5+3) \bmod 16 = 5$

$R_{11} = (5*5+3) \bmod 16 = 12$

$R_{12} = (12*5+3) \bmod 16 = 15$

$R_{13} = (15*5+3) \bmod 16 = 14$

$R_{14} = (14*5+3) \bmod 16 = 9$

$R_{15} = (9*5+3) \bmod 16 = 0$

$R_{16} = (0*5+3) \bmod 16 = 3$

Hasil yang diperoleh menunjukkan bahwa nilai awal kembali muncul pada R_{16}. Sehingga suku-suku barisan akan berulang untuk setiap kelipatan 16. Barisan bilangan pseudorandom hasil percobaan adalah :

3, 2, 13, 4, 7, 6, 1, 8, 11, 10, 5, 12, 15, 14, 9, 0.

Secara umum, untuk memperoleh barisan bilangan random berdistribusi uniform seperti pada percobaan Fuller, maka syarat yang harus dipenuhi adalah

1. Banyaknya bilangan yang dibangkitkan adalah N = m-1, dengan N dimulai dari 0 (nol). Jadi i = 0, 1, 2, ... m-1.
2. Modulo sebaiknya merupakan pangkat dari 2.

$m = 2^k$, k = 2,3,4,....,b.

b = maksimum bit komputer.

Barisan bilangan pseudorandom yang dihasilkan masih dalam bentuk bilangan bulat dalam interval [0,m], dengan demikian harus ditransformasikan ke dalam interval [0,1] untuk proses generalisasi.

Fungsi transformasinya :

$$U_i = \frac{R_i}{m} \tag{2.43}$$

dimana R_i = Bilangan random ke-i, i = 0,1,2,..., m-1

m = modulo

U_i = Bilangan random ke-i pada interval [0,1]

Berbagai konstanta persamaan pembangkitan (tetapan) yang telah teruji sehingga dapat dikatakan menghasilkan barisan bilangan pseudorandom yang berdistribusi uniform, antara lain :

1. Knuth (1971)

   p = 3141592653

   c = 2718281829

   m = 2^{35}

   R_0 = 0

2. P. Siagian (1987)

   p = 100.003

   m = 10^{10}

   R_0 = 123.456.789

tetapan ini mempunyai daur $R_n = R_0$ pada $n = 5 \times 10^8$

2.5. Bahasa Pemrograman Visual Basic 6.0

Di dalam dunia pemrograman, telah dikenal istilah pemrograman terstruktur. Contohnya adalah bahasa Pascal, C, Delphi dan laian-lain. Pemrograman tersruktur mempunyai pengertian bahwa suatu program yang didesain untuk menyelesaikan masalah yang besar, disusun dengan cara memecahkannya menjadi beberapa bagian, sehingga setiap bagian menangani satu masalah kecil dan kemudian mengintegrasikan semua bagian bagian tersebut sesuai urutan algoritmanya.

Perkembangan selanjutnya dari *structured programming* adalah *Object Oriented Programming* (OOP) atau pemrograman beorientasi obyek. Seperti halnya pemrograman terstruktur, OOP menyederhanakan permasalahan menjadi obyek-obyek yang terpisah. Setiap obyek mempunyai karakteristik (*properties*) tersendiri, tingkah laku (*method*) dan kejadian (*event*) yang menjalankan *event*-nya. Obyek-obyek disini dapat diartikan benda, manusia, tempat atau semua hal yang mempunyai atribut dan tingkah laku. Misal sebagai obyek adalah manusia, atributnya adalah nama, alamat, tanggal lahir dan metode-metode nya adalah berjalan, makan, diam dan lain-lain. Kejadian (*event*) adalah keadaan yang dapat dikenakan pada obyek tersebut.

Visual Basic 6.0 merupakan salah satu bahasa pemrograman berorientasi obyek. Visual Basic memuat *propertiy, methods* dan *event* dari suatu obyek yang telah didefinisikan. Namun yang akan diulas disini adalah cara penulisan kode (struktur program) dari Visual Basic 6.0.

Elemen penting dalam bahasa Visual Basic 6.0 adalah Forms, module dan class module.

Forms

Forms adalah suatu obyek yang berfungsi sebagai median komunikasi antara program dengan pemakai. Pada forms ini dapat diletakkan alat alat kontrol seperti tombol (command button), tulisan (textbox dan label), kaotak gambar (picturebox) dan lain-lain.

Class Module

Tempat pendefinisian obyek-obyek baru selain yang telah disediakan oleh Visual Basic 6.0.

Module

Tempat penulisan program (modul-modul) yang berisi subprogram (sub dan function). Meskipun demikian kode kode program untuk pengendali event (*event handler*) terletak dimana obyek tersebut ditempatkan.

Penulisan kode

Kode-kode program Visual Basic hampir sama dengan bahasa Basic versi DOS, dengan perluasan reserved word dan aturan-aturan penulisan atau sintaksis.

Untuk mendeklarasikan suatu variabel aturannya sebagai berikut :

Dim *nama_variabel* As *Type_variabel*

Private (Public) *nama_variabel* As *Type_variabel*

Tipe-tipe varibel didalam Visual Basic adalah Integer, Long, Single, Double, String, Byte, Decimal, Object, Variant, Array, Date, Currency dan Boolean.

Penulisan subprogram-subprogram dalam VB dikenal dengan istilah Sub dan Function.

```
Sub nama_subprogram (daftar parameter)
Deklarasi variabel
—
—      }   statement
—
End sub
```

```
Function nama_function (daftar parameter)
Deklarasi variabel
—
—      }   statement
—
Endfunction
```

Daftar parameter berisi parameter apa saja yang dilewatkan kedalam subprogram tersebut. Kode untuk pengendali event biasanya didahului kata *private*.

```
Private Sub command1_click ()
Deklarasi variabel
—
—      }   statement
—
End sub
```

Kode diatas berarti kode untuk obyek tombol (command1) ketika ditekan (click).

```
Private Sub Form1_Load ()
Deklarasi variabel
—
—        } statement
—
End sub
```

Kode ini dijalankan ketika sebuah form di-*load*.

Sementara itu program utama (main program) biasanya ditulis dalam subprogram yang diberi nama Main

```
Sub Main()
—
—
End Sub
```

Kemampuan yang dimiliki oleh Visual Basic 6.0, seperti halnya bahasa berorientasi obyek yang lain, adalah melakukan *link* dan *Embedding* atau OLE dan kontrol ActiveX sehingga suatu program dapat mengakses program program lain meskipun tidak ada hubungan atau kaitannya dengan program yang sedang dijalankan. Selain itu juga dapat digunakan untuk pemrograman basis data.