BAB II

TEORI PENUNJANG

Dalam bab ini dijelaskan beberapa teori yang dapat menunjang materi pada bab III antara lain teori himpunan dan fungsi, matriks dan transformasi, topologi metrik, geometri fraktal, algoritma serta pemrograman Turbo Pascal 7.0.

2.1. Himpunan dan Fungsi

Jika A menotasikan ***himpunan*** dan a ada dalam A, maka dikatakan a adalah ***anggota*** A, dan dinotasikan dengan $a \in A$. Jika a tidak berada di A, ditulis $a \notin A$.

Jika A dan B adalah himpunan, maka dikatakan A adalah ***himpunan bagian*** dari B apabila setiap anggota A juga merupakan anggota B, ditulis $A \subseteq B$.

Definisi 2. 1. 1. ***Irisan*** dari himpunan A dan B, yang dinotasikan dengan $A \cap B$, adalah himpunan semua objek yang menjadi anggota A dan B. ♦

Definisi 2. 1. 2. ***Gabungan*** dari himpunan A dan B, dinotasikan dengan $A \cup B$, adalah himpunan dari semua objek yang menjadi anggota A atau B. ♦

Definisi 2. 1. 3. ***Komplemen dari himpunan B relatif ke himpunan A***, yang dinotasikan dengan $A - B$, adalah himpunan dari semua objek yang menjadi anggota A tetapi bukan anggota B. ♦

Definisi 2. 1. 4. ***Perkalian Cartesian*** dari himpunan A dan B adalah himpunan dari semua pasangan terorde $<a, b>$ dengan $a \in A$ dan $b \in B$. Perkalian Cartesian dari himpunan A dan B dinotasikan dengan $A \times B$. ♦

Teori himpunan yang dibahas dalam sub bab ini hanyalah beberapa operasi himpunan yang mendasar yang digunakan dalam pembahasan berikutnya. Masih banyak lagi operasi himpunan lain yang tidak ditunjukkan di sini.

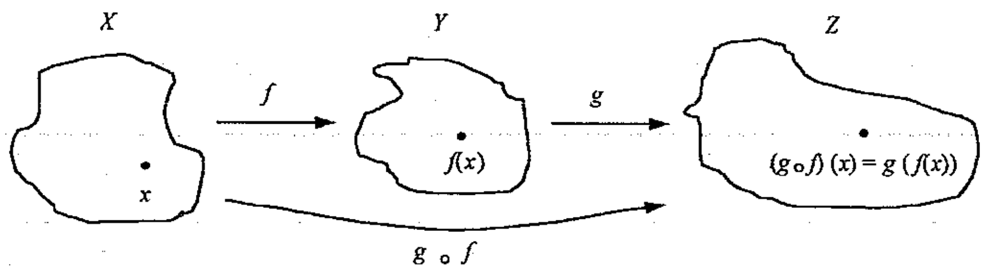
Definisi 2. 1. 5. ***Fungsi*** f dari himpunan X ke himpunan Y yang dinotasikan oleh $f: X \to Y$ adalah suatu aturan yang menentukan setiap $x \in X$ dengan $y = f(x) \in Y$. X disebut ***domain*** dari f, dan Y disebut ***kodomain*** dari f. ♦

$f(x)$ disebut ***nilai*** dari f pada x atau ***bayangan*** dari x di bawah f, dan ***daerah hasil*** dari f adalah $\{ f(x) \mid x \in X \}$. Misal $A \subseteq X$, himpunan $f(A)$ didefinisikan oleh

$$f(A) = \{ y \in Y \mid \text{ada } x \in A \text{ sedemikian sehingga } f(x) = y \} \qquad (2.\ 1.\ 1)$$

sehingga daerah hasil dari f dapat dituliskan dengan $f(X)$.

Definisi 2. 1. 6. Misal $f: X \to Y$ dan $g: Y \to Z$ keduanya merupakan fungsi. Fungsi $g \circ f: X \to Z$ yang didefinisikan dengan $(g \circ f)(x) = g(f(x))$ disebut dengan ***komposisi*** dari f dan g. ♦



Gambar 2. 1. 1. Komposisi dari fungsi f dan g.

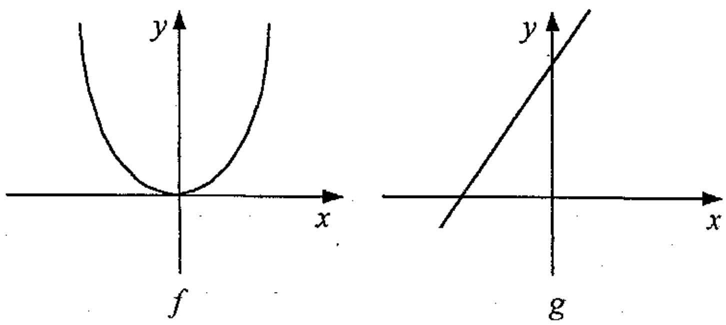
Definisi 2. 1. 7. Fungsi $f : X \to Y$ dikatakan ***injektif*** atau ***satu-satu*** yaitu untuk semua $x_1, x_2 \in X$ berlaku: jika $f(x_1) = f(x_2)$, maka $x_1 = x_2$. ♦

Definisi 2. 1. 8. Fungsi $f : X \to Y$ dikatakan ***surjektif*** atau ***pemetaan X onto Y*** yaitu apabila untuk setiap $y \in Y$, ada $x \in X$ sedemikian sehingga $f(x) = y$. ♦

Definisi 2. 1. 9. Fungsi $f : X \to Y$ dikatakan ***bijektif*** atau ***berkorespondensi satu-satu*** apabila f adalah injektif dan surjektif. ♦

Contoh 2. 1. 1

Fungsi $f : \mathbb{R} \to \mathbb{R}$ yang didefinisikan dengan $f(x) = x^2$ untuk tiap $x \in \mathbb{R}$ bukan fungsi injektif karena $f(-1) = f(1)$, dan f juga bukan fungsi surjektif karena untuk setiap $f(x) < 0$ maka $x \notin \mathbb{R}$. Sedangkan fungsi $g : \mathbb{R} \to \mathbb{R}$ yang didefinisikan dengan $g(x) = 2x + 3$ merupakan fungsi yang injektif karena jika $g(x) = g(y)$, maka $2x + 3 = 2y + 3$ sehingga didapatkan $x = y$, dan g juga merupakan fungsi yang surjektif karena untuk setiap $y \in \mathbb{R}$, maka terdapat $x \in \mathbb{R}$.



Gambar 2. 1. 2. f bukan fungsi yang injektif maupun surjektif, sedangkan g merupakan fungsi yang injektif dan surjektif.

Jika $f : X \to Y$ bijektif, maka terdapat fungsi $f^{-1} : Y \to X$ yang disebut ***invers*** dari f. Fungsi f^{-1} didefinisikan dengan menentukan $f^{-1}(y)$ sebagai elemen unik $x \in X$ sedemikian sehingga $f(x) = y$.

Definisi 2. 1. 10. Himpunan A dikatakan ***terhingga*** (*finite*) jika himpunan itu kosong atau jika ada bijeksi dengan domain A dan kodomain suatu bagian bilangan asli $\mathbb{N}$. Selain itu, maka himpunan A dikatakan ***tak terhingga*** (*infinite*).♦

Definisi 2. 1. 11. Misal $\mathcal{A}$ koleksi himpunan yang tidak kosong. ***Gabungan*** dari anggota-anggota $\mathcal{A}$ didefinisikan oleh $\bigcup\{A \mid A \in \mathcal{A}\} = \{x \mid x \in A$ untuk suatu $A \in \mathcal{A}\}$, sedangkan ***irisan*** dari anggota-anggota $\mathcal{A}$ didefinisikan oleh $\bigcap\{A \mid A \in \mathcal{A}\} = \{x \mid x \in A$ untuk semua $A \in \mathcal{A}\}$. ♦

Definisi 2. 1. 12. Misal $\mathcal{A}$ koleksi himpunan yang tidak kosong. ***Fungsi terindeks*** untuk $\mathcal{A}$ adalah fungsi $f : \Lambda \to \mathcal{A}$. Himpunan Λ disebut ***himpunan indeks*** untuk $\mathcal{A}$, dan $\mathcal{A}$ bersama fungsi terindeks f disebut ***keluarga himpunan terindeks***. ♦

Jika $\alpha \in \Lambda$, dan A_α menotasikan $f(\alpha)$, maka keluarga himpunan terindeks dinotasikan dengan $\{A_\alpha \mid \alpha \in \Lambda\}$. Jadi $\bigcup\{A \mid A \in \mathcal{A}\}$ dapat ditulis $\bigcup\{A_\alpha \mid \alpha \in \Lambda\}$ atau $\bigcup_{\alpha\in\Lambda} A_\alpha$, dan $\bigcap\{A \mid A \in \mathcal{A}\}$ dapat ditulis $\bigcap\{A_\alpha \mid \alpha \in \Lambda\}$ atau $\bigcap_{\alpha\in\Lambda} A_\alpha$.

Misal $I_n = \{i \in \mathbb{N} \mid i \le n\}$ untuk tiap $n \in \mathbb{N}$. Suatu ***tupel-n*** dari anggota-anggota A adalah fungsi $f : I_n \to A$. Jika $\{A_i \mid i \in I_n\}$ adalah koleksi dari n himpunan, maka $\bigcup\{A_i \mid i \in I_n\}$ dinotasikan dengan $\bigcup_{i=1}^{n} X_i$, dan $\bigcap\{A_i \mid i \in I_n\}$ dengan $\bigcap_{i=1}^{n} X_i$.

Definisi 2. 1. 13. Suatu ***barisan*** dalam himpunan X adalah fungsi dengan domain $\mathbb{N}$ dan kodomain X. Notasi untuk barisan adalah $\{x_n\}$, dengan $x \in X$ dan $n \in \mathbb{N}$. ♦

Definisi 2. 1. 14. Misal $S \subseteq \mathbb{R}$. $u \in \mathbb{R}$ disebut ***batas atas*** dari S jika $s \leq u$ untuk semua $s \in S$, dan $w \in \mathbb{R}$ disebut ***batas bawah*** dari S jika $w \leq s$ untuk semua $s \in S$. ♦

S dikatakan ***terbatas di atas*** jika mempunyai batas atas, dan dikatakan ***terbatas di bawah*** jika mempunyai batas bawah.

Definisi 2. 1. 15. Misal $S \subseteq \mathbb{R}$. Jika S terbatas di atas, maka batas atas dari S dikatakan ***supremum*** dari S jika batas atas itu lebih kecil dari batas atas dari S yang lain. Jika S terbatas di bawah, maka batas bawah S dikatakan ***infimum*** dari S jika batas bawah itu lebih besar dari batas bawah dari S yang lain. ♦

2.2. Matriks dan Transformasi

Definisi 2. 2. 1. ***Matriks*** adalah himpunan skalar yang disusun secara empat persegi panjang (menurut baris-baris dan kolom-kolom). Skalar-skalar itu disebut ***elemen*** atau ***entri*** matriks. Sedangkan untuk batasnya digunakan $\begin{bmatrix} & \end{bmatrix}$. ♦

Suatu matriks berukuran m baris dan n kolom:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

dapat dinotasikan dengan $A_{m \times n} = [a_{ij}]$, dengan $i = 1, 2, \ldots, m$ dan $j = 1, 2, \ldots, n$.

Definisi 2. 2. 2. ***Matriks kuadrat*** berukuran n adalah suatu matriks yang banyaknya baris sama dengan banyaknya kolom yaitu n. ♦

Definisi 2. 2. 3. Jika A dan B adalah dua matriks yang ukurannya sama, maka ***jumlah*** A dan B, yang dinotasikan dengan $A + B$, adalah matriks yang didapatkan dengan menambahkan entri yang seletak di dalam kedua matriks tersebut. ♦

Definisi 2. 2. 4. Jika A adalah suatu matriks dan $k \in \mathbb{R}$ adalah suatu skalar, maka ***hasil kali*** skalar k dengan matriks A adalah matriks yang didapatkan dengan mengalikan k dengan setiap entri dari A. ♦

Definisi 2. 2. 5. Jika $A_{p \times q} = [a_{ij}]$ dan $B_{q \times r} = [b_{jk}]$, maka ***hasil kali*** A dengan B, yang dinotasikan dengan AB, adalah suatu matriks $C_{p \times r} = [c_{ik}]$ dengan $c_{ik} = a_{i1}\, b_{1k} + a_{i2}\, b_{2k} + \ldots + a_{ij}\, b_{jk}$, untuk setiap $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, q$, dan $k = 1, 2, \ldots, r$. ♦

Definisi 2. 2. 6. ***Ruang vektor*** adalah himpunan V (yang anggota-anggotanya disebut ***vektor***) yang dilengkapi dengan dua operasi biner, yaitu operasi penjumlahan vektor (*vector addition*) dan operasi pergandaan skalar (*scalar multiplication*). ♦

Jika $\mathbf{v}, \mathbf{w} \in V$, maka terdapat $\mathbf{v} + \mathbf{w} \in V$ yang disebut vektor jumlah dari $\mathbf{v}$ dan $\mathbf{w}$. Jika $k \in \mathbb{R}$ dan $\mathbf{v} \in V$, maka terdapat $k\mathbf{v} \in V$ yang disebut pergandaan dari k dan $\mathbf{v}$. Operasi penjumlahan dan pergandaan vektor tersebut memenuhi ***aksioma ruang vektor*** berikut:

1. $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$ untuk semua $\mathbf{v}, \mathbf{w} \in V$;
2. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ untuk semua $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$;

3. $\exists\ \mathbf{0} \in V$ sedemikian sehingga $\mathbf{0} + \mathbf{v} = \mathbf{v} + \mathbf{0} = \mathbf{v}$ untuk semua $\mathbf{v} \in V$;
4. $\forall\ \mathbf{v} \in V, \exists\ \text{-}\mathbf{v} \in V$ sedemikian sehingga $\mathbf{v} + (-\mathbf{v}) = (-\mathbf{v}) + \mathbf{v} = \mathbf{0}$;
5. $\mathbf{1}\cdot\mathbf{v} = \mathbf{v}\cdot\mathbf{1} = \mathbf{v}$ untuk semua $\mathbf{v} \in V$;
6. $k\,(l\mathbf{v}) = (kl)\mathbf{v}$ untuk semua $k, l \in \mathbb{R}$ dan $\mathbf{v} \in V$;
7. $k\,(\mathbf{u} + \mathbf{v}) = k\,\mathbf{u} + k\,\mathbf{v}$ dan $(k + l)\,\mathbf{v} = k\,\mathbf{v} + l\,\mathbf{v}$ untuk semua $k, l \in \mathbb{R}$ dan $\mathbf{u}, \mathbf{v} \in V$;

Definisi 2. 2. 7. Jika $F : V \to W$ adalah sebuah fungsi dari ruang vektor V ke dalam ruang vektor W, maka F dinamakan ***transformasi linier*** jika

(i) $F\,(\mathbf{u} + \mathbf{v}) = F\,(\mathbf{u}) + F\,(\mathbf{v})$ untuk semua $\mathbf{u}, \mathbf{v} \in V$.

(ii) $F\,(k\mathbf{u}) = k\,F\,(\mathbf{u})$ untuk semua $\mathbf{u} \in V$ dan semua $k \in \mathbb{R}$. ♦

Definisi 2. 2. 8. ***Transformasi matriks*** adalah transformasi linier $F : \mathbb{R}^n \to \mathbb{R}^m$ yang didefinisikan dengan $F(\mathbf{x}) = A\mathbf{x}$, dengan $\mathbf{x}$ adalah vektor dan $A_{m \times n}$ menotasikan matriks untuk vektor di dalam $\mathbb{R}^m$ dan $\mathbb{R}^n$. ♦

Contoh 2. 2. 1

Misal θ adalah sebuah sudut tetap dan $F : \mathbb{R}^2 \to \mathbb{R}^2$ adalah perkalian oleh matriks

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad (2.\ 2.\ 1)$$

Jika $\mathbf{v} \in \mathbb{R}^2$, maka

$$F\,(\mathbf{v}) = A\mathbf{v} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix} \qquad (2.\ 2.\ 2)$$

Secara geometris, $F\,(\mathbf{v})$ adalah vektor yang dihasilkan jika $\mathbf{v}$ dirotasikan oleh sudut θ. Untuk melihat F linier, misal ϕ adalah sudut antara $\mathbf{v}$ dan sumbu-x positip dan

$$\mathbf{v}' = \begin{bmatrix} x' \\ y' \end{bmatrix} \tag{2. 2. 3}$$

adalah vektor yang dihasilkan dari rotasi **v** oleh sudut θ (Gambar 2.2.1), maka akan ditunjukkan bahwa $\mathbf{v}' = F(\mathbf{v})$. Jika r menyatakan panjang dari **v**, maka

$$x = r\cos\phi, \quad y = r\sin\phi \tag{2. 2. 4}$$

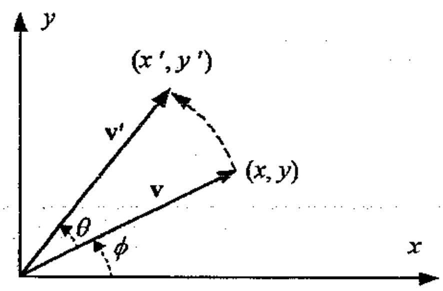
Demikian juga, karena **v′** mempunyai panjang yang sama dengan **v**, maka

$$x' = r\cos(\theta + \phi), \quad y' = r\sin(\theta + \phi) \tag{2. 2. 5}$$

Sehingga

$$\begin{aligned}
\mathbf{v}' &= \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} r\cos(\theta+\phi) \\ r\sin(\theta+\phi) \end{bmatrix} \\
&= \begin{bmatrix} r\cos\theta\cos\phi - r\sin\theta\sin\phi \\ r\sin\theta\cos\phi + r\cos\theta\sin\phi \end{bmatrix} \\
&= \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= A\mathbf{v} = F(\mathbf{v})
\end{aligned}$$

Transformasi linier di dalam contoh ini dinamakan ***rotasi dari*** $\mathbb{R}^2$ ***melalui sudut*** θ.



Gambar 2. 2. 1. Rotasi (x, y) oleh sudut θ .

2.3. Topologi Metrik

Definisi 2. 3. 1. ***Metrik*** pada sebarang himpunan X adalah fungsi $d : X \times X \to \mathbb{R}$ yang memenuhi aksioma berikut:

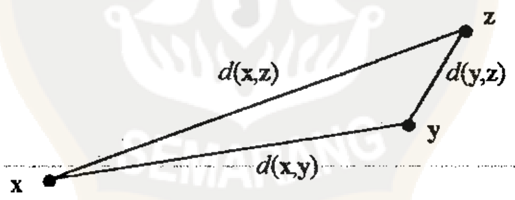
(**D**1) $0 \leq d(\mathbf{x}, \mathbf{y})$ untuk semua $\mathbf{x}, \mathbf{y} \in X$, dengan kesamaan jika dan hanya jika $\mathbf{x} = \mathbf{y}$.

(**D**2) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ untuk semua $\mathbf{x}, \mathbf{y} \in X$, dan

(**D**3) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ untuk semua $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$.

$d(\mathbf{x}, \mathbf{y})$ disebut ***jarak*** dari **x** ke y, dan pasangan (X, d) disebut ***ruang metrik***. ♦

Aksioma (**D**3) pada metrik adalah abstraksi dari fakta bahwa jarak dari satu sisi segitiga adalah kurang dari atau sama dengan jumlahan dari jarak dua sisi lainnya. Hal ini disebut ***ketidaksamaan segitiga***.



Gambar 2. 3. 1. Ketidaksamaan segitiga: $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$

Contoh 2. 3. 1

Fungsi $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ yang didefinisikan oleh $d(x, y) = |x - y|$ memenuhi **D**1-**D**3, sehingga fungsi d adalah metrik pada $\mathbb{R}$, dan disebut ***metrik biasa*** untuk $\mathbb{R}$.

Definisi 2. 3. 2. Untuk setiap $n \geq 1$, ***ruang-n riil*** yang dinotasikan dengan $\mathbb{R}^n$ adalah himpunan semua $\mathbb{R} \times \ldots \times \mathbb{R}$ atau tupel-n bilangan riil $(x_1, x_2, \ldots, x_n)$. ♦

Ruang-n riil $\mathbb{R}^n$ juga mempunyai suatu metrik atau fungsi jarak. Misal $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, maka ***metrik Euclidis*** adalah fungsi d: $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ yang didefinisikan oleh

$$d(\mathbf{x}, \mathbf{y}) = \left(\textstyle\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2} \qquad (2.\ 3.\ 1)$$

Definisi 2. 3. 3. ***Ruang-n Euclidis*** yang dinotasikan dengan $\mathbb{E}^n$ adalah ruang-n riil $\mathbb{R}^n$ yang dilengkapi dengan metrik Euclidis d. ♦

Definisi 2. 3. 4. Untuk tiap $\mathbf{x} \in X$ dan bilangan riil $r > 0$, himpunan titik-titik dalam X yang berjarak kurang dari r dari titik $\mathbf{x}$ yang tertentu disebut ***bola buka*** sekitar $\mathbf{x}$ dengan jari-jari r, dan ditulis $B_d(\mathbf{x}, r) = \{\mathbf{y} \in X \mid d(\mathbf{x}, \mathbf{y}) < r\}$. Himpunan titik-titik dalam X yang berjarak kurang dari atau sama dengan r dari titik $\mathbf{x}$ yang tertentu disebut ***bola tutup*** sekitar $\mathbf{x}$, dan ditulis $D_d(\mathbf{x}, r) = \{ \mathbf{y} \in X \mid d(\mathbf{x}, \mathbf{y}) \le r \}$. ♦

Definisi 2. 3. 5. Misal (X, d) ruang metrik, dan misal $\varepsilon > 0$. $U \subseteq X$ adalah ***buka*** jika untuk setiap $\mathbf{x} \in U$, ada bola buka $B_d(\mathbf{x}, \varepsilon)$ sedemikian sehingga $B_d(\mathbf{x}, \varepsilon) \subseteq U$. Sedangkan $F \subseteq X$ adalah ***tertutup*** jika $X - F$ adalah buka. ♦

Definisi 2. 3. 6. Misal (X, d) dan (Y, ρ) ruang metrik. Fungsi $F : X \to Y$ ***kontinu pada suatu titik*** $\mathbf{a} \in X$ artinya, untuk setiap $\varepsilon > 0$ terdapat $\delta > 0$ sedemikian sehingga jika $\mathbf{x} \in X$ dan $d(\mathbf{a}, \mathbf{x}) < \delta$, maka $\rho(F(\mathbf{a}), F(\mathbf{x})) < \varepsilon$. Fungsi $F : X \to Y$ adalah ***kontinu*** jika fungsi tersebut kontinu pada setiap titik dari X. ♦

Contoh 2. 3. 2

Misal (X, d) dan (Y, ρ) ruang metrik. Fungsi $I : X \to X$ yang didefinisikan oleh $I(\mathbf{x}) = \mathbf{x}$ dan $C: X \to Y$ yang didefinisikan oleh $C(\mathbf{x}) = \mathbf{y}$ untuk semua $\mathbf{x} \in X$ adalah kontinu.

Untuk melihatnya, misal $\mathbf{a} \in X$ dan $\varepsilon > 0$. Misal $\delta = \varepsilon$. Jika $\mathbf{x} \in X$ dan $d(\mathbf{a}, \mathbf{x}) < \delta$, maka $\rho(I(\mathbf{a}), I(\mathbf{x})) = \rho(\mathbf{a}, \mathbf{x}) < \varepsilon$. Sehingga I kontinu. Kemudian misal δ sebarang bilangan positip. Jika $\mathbf{x} \in X$ dan $d(\mathbf{a}, \mathbf{x}) < \delta$, maka $\rho(C(\mathbf{a}), C(\mathbf{x})) = \rho(\mathbf{y}, \mathbf{y}) = 0 < \varepsilon$. Sehingga C juga kontinu.

Definisi 2. 3. 7. Misal (X, d) ruang metrik. Koleksi $\mathcal{A}$ disebut ***sampul*** dari $K \subseteq X$ jika dipenuhi $K \subseteq \bigcup_{A \in \mathcal{A}} A$, dan $\mathcal{A}$ disebut ***sampul buka*** dari A jika setiap $A \in \mathcal{A}$ adalah himpunan buka. $K \subseteq X$ adalah ***kompak*** jika setiap sampul buka dari K mempunyai sampul bagian yang terhingga. ♦

Contoh 2. 3. 3

Misal $K = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\} \subseteq \mathbb{R}^n$. Jika $\mathcal{G} = \{G_\alpha\}$ adalah koleksi dari himpunan buka di $\mathbb{R}^n$, dan untuk tiap $\mathbf{x} \in K$ berlaku $\mathbf{x} \in G_\alpha$, maka ada paling banyak m himpunan bagian dari $\mathcal{G}$ yang mana gabungannya memuat K. Oleh karena itu $K \subseteq \mathbb{R}^n$ kompak.

Definisi 2. 3. 8. Barisan $\{\mathbf{x}_n\}$ dari ruang metrik (X, d) ***konvergen*** ke $\mathbf{x} \in X$, jika $\{d(\mathbf{x}_n, \mathbf{x})\}$ konvergen ke 0. $\mathbf{x}$ disebut ***limit*** barisan $\{\mathbf{x}_n\}$, dan ditulis $\mathbf{x} = \lim_{n \to \infty} \mathbf{x}_n$. ♦

Definisi 2. 3. 9. Barisan $\{\mathbf{x}_n\}$ dari ruang metrik (X, d) adalah ***barisan Cauchy*** jika untuk tiap $\varepsilon > 0$ terdapat $k \geq 1$ ($k \in \mathbb{N}$) sedemikian sehingga jika $m, n \geq k$ maka $d(\mathbf{x}_m, \mathbf{x}_n) < \varepsilon$. ♦

Definisi 2. 3. 10. Ruang metrik (X, d) beserta metrik d disebut ***lengkap*** jika setiap barisan Cauchy dalam (X, d) konvergen di (X, d). ♦

Teorema 2. 3. 1. Bayangan kontinu dari ruang yang kompak juga kompak. ♦

Bukti. Misal (X, d) ruang metrik kompak, misal (Y, ρ) ruang metrik, misal $\mathcal{U}$ sampul buka dari Y, dan misal $F : X \rightarrow Y$ fungsi kontinu. Maka $\mathcal{U}' = \{F^{-1}(U) \mid U \in \mathcal{U}\}$ adalah sampul buka dari X. Karena X kompak, $\mathcal{U}'$ mempunyai suatu sampul bagian terhingga $\mathcal{U}''$. Sehingga $\{U \in \mathcal{U} \mid F^{-1}(U) \in \mathcal{U}''\}$ adalah koleksi bagian terhingga dari $\mathcal{U}$ yang menyampuli Y. Karenanya Y kompak. □

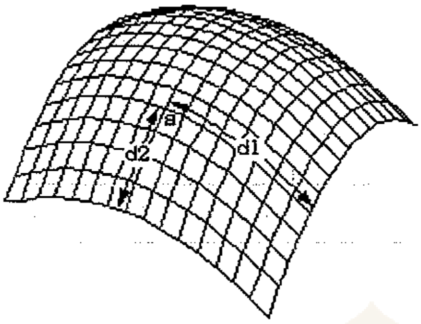
2.4. Geometri Fraktal

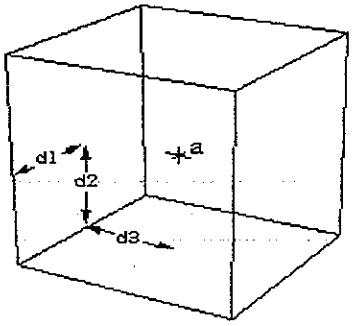
Salah satu faktor yang membedakan geometri Euclidis dengan geometri fraktal adalah dimensinya. Dimensi dalam geometri Euclidis dinyatakan dengan bilangan bulat, misalnya 1 untuk garis, 2 untuk persegi, dan 3 untuk kubus, sedangkan geometri fraktal mempunyai dimensi pecahan.

Dimensi dapat dideskripsikan secara intuitif dan matematis. Secara intuitif dimensi dideskripsikan sebagai besarnya bilangan yang diperlukan untuk mendefinisikan sebarang titik secara unik. Misalnya dikatakan bahwa garis adalah satu dimensi karena hanya membutuhkan satu bilangan untuk mendefinisikan sebarang titik dalam garis itu secara unik.

Gambar 2. 4. 1. Sebarang titik a pada dimensi satu didefinisikan dengan satu bilangan.

Bidang adalah dua dimensi karena untuk mendefinisikan sebarang titik pada permukaannya dibutuhkan dua bilangan. Sedangkan suatu objek padat adalah tiga dimensi karena diperlukan tiga bilangan untuk mendefinisikan sebarang titik pada objek itu secara unik.



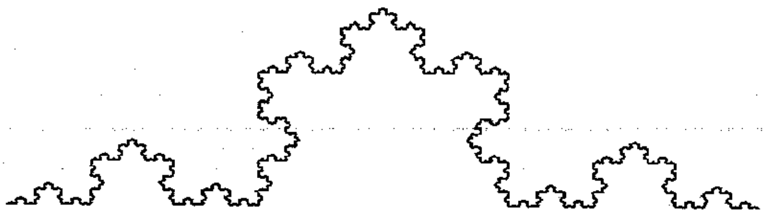


Gambar 2. 4. 2. Sebarang titik a pada bidang dimensi dua dapat didefinisikan dengan dua bilangan, sedangkan dalam dimensi tiga dengan tiga bilangan.

Secara matematis konsep dimensi ini dapat dideskripsikan sebagai berikut. Jika mengukur objek dua dimensi, maka luasnya meningkat dengan skala persegi, sedangkan pada objek tiga dimensi, volume meningkat dengan skala kubik. Pengukuran objek memberi bayangan mengenai dimensi (dim) yang tergantung pada bagaimana perubahan ukuran (s) sebagai sebuah fungsi dari skala linear (l), yaitu:

$$\dim = \frac{\log(s)}{\log(l)} \tag{2. 4. 1}$$

Jika formula (2.4.1) untuk dimensi digunakan, maka terlihat bahwa formula itu tidak hanya menghasilkan bilangan bulat. Geometri dengan dimensi yang bukan bilangan bulat inilah yang disebut dengan geometri fraktal.

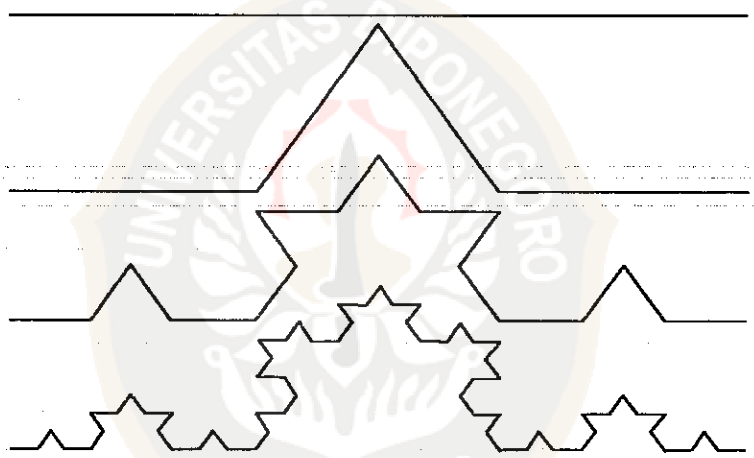


Gambar 2. 4. 3. Gambar fraktal pantai Von Koch

Salah satu contoh gambar fraktal yang sederhana adalah garis pantai Von Koch (Gambar 2.4.3). Metode pembentukannya dimulai dari sepotong garis, kemudian menggantinya dengan 4 bagian garis (Gambar 2.4.4).

Gambar 2. 4. 4. Pembangkit untuk pembuatan garis pantai Von Koch.

Beberapa iterasi pertama dari metode itu di ditunjukkan dari Gambar 2.4.5.



Gambar 2. 4. 5. Tiga iterasi pertama pembentukan fraktal Von Koch.

Perbedaan lain antara geometri fraktal dan geometri Euclidis yaitu dalam hal similaritas. Jika bentuk Euclidis diperbesar, seperti misalnya tepi dari suatu lingkaran, maka perbesaran itu memperlihatkan bentuk yang berbeda, yaitu garis lurus. Sedangkan jika gambar fraktal diperbesar terus menerus, maka detailnya sama dengan dirinya sendiri (*self similar*). Sifat *self similar* inilah yang menonjol dari geometri fraktal dan digunakan oleh Mandelbrot untuk mendefinisikan fraktal.

Definisi 2. 4. 1. ***Fraktal*** adalah suatu bentuk objek yang masing-masing bagian dari objek itu merupakan merupakan tiruan dari bagian keseluruhannya. ♦

Geometri fraktal dan geometri Euclidis juga mempunyai perbedaan dalam hal penyajian posisi objek. Bentuk-bentuk pada geometri Euclidis dapat dijelaskan posisinya secara intuitif dengan bilangan unik. Sedangkan pada geometri fraktal, meskipun misalnya objek fraktal pada Gambar 2.4.3 memperlihatkan kurva dengan titik awal dan akhir, namun tidak ada formula aljabar untuk mendeskripsikan titik-titik pada kurva yang memungkinkan untuk secara unik menyebutkan satu persatu sebarang posisi sepanjang kurva seperti pada kurva Euclidis dimensi satu.

Tabel 2. 4. 1. Perbedaan geometri fraktal dan geometri Euclidis.

Fraktal	Euclidis
hasil modern	tradisional
tidak ada ukuran/skala spesifik	berdasarkan ukuran/skala karakteristik
disediakan untuk geometri alam	digunakan untuk objek buatan manusia
didefinisikan dengan algoritma	didefinisikan dengan formula sederhana

2.5. Algoritma

Definisi 2. 5. 1. ***Algoritma*** adalah himpunan instruksi yang menjelaskan langkah-langkah yang teratur dan berhingga untuk menyelesaikan suatu masalah. ♦

Kriteria algoritma yang baik adalah:

1. *Adanya input*: suatu algoritma sekurang-kurangnya mempunyai sejumlah satu atau lebih input yang disuplai secara eksternal.
2. *Adanya output*: input suatu algoritma harus memberikan output sekurang-kurangnya sejumlah satu input.
3. *Kepastian*: tiap instruksi dalam algoritma harus jelas dan tidak berarti ganda.

4. *Keberhinggaan*: apabila instruksi dalam algoritma ditelusuri, maka algoritma berhenti setelah sejumlah berhingga langkah-langkah.
5. *Keefektifan*: tiap instruksi harus langsung mengarah ke sasaran yang dituju.

Ada beberapa cara untuk menyatakan algoritma, yaitu:

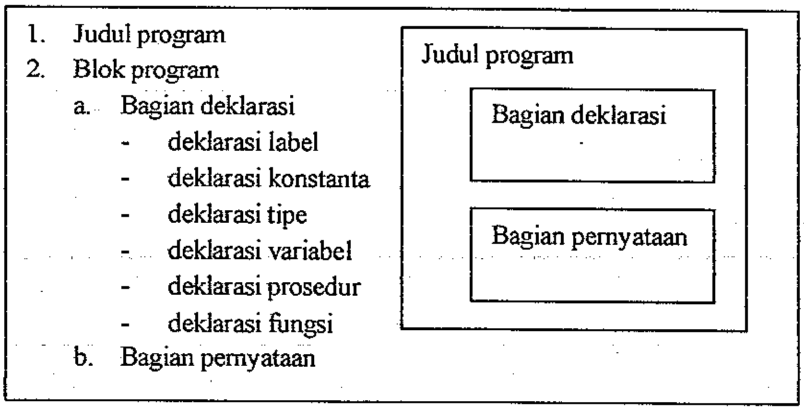
1. *Dengan bahasa alamiah*: algoritma ditulis dengan ungkapan sehari-hari.
2. *Dengan sandi semu (Pseudocode)*: algoritma ditulis dengan menggunakan perintah bahasa pemrograman tertentu ditambah dengan bahasa alamiah.
3. *Dengan bagan alir (flowchart)*: algoritma ditulis dalam bentuk simbol/ diagram.

2.6. Pemrograman Turbo Pascal 7.0

Pascal adalah bahasa tingkat tinggi yang orientasinya pada segala tujuan, dirancang oleh Niklaus Wirth dari Technical University Switzerland. Turbo Pascal adalah salah satu versi dari Pascal selain UCSD Pascal, MS-Pascal, Apple Pascal, dan lain sebagainya. Hal-hal yang perlu diketahui dalam Turbo Pascal diantaranya:

2.6.1. Struktur Program Pascal

Secara ringkas, struktur program Pascal dapat dilihat dalam Gambar 2.6.1.



Gambar 2. 6. 1. Struktur program Pascal.

2.6.2. Tipe Data

Dalam Pascal semua variabel yang dipakai harus sudah ditentukan tipe datanya. Penentuan tipe data suatu variabel, berarti juga penentuan batasan nilai variabel tersebut dan jenis operasi yang bisa dikerjakan atas variabel tersebut.

Bentuk umum dari deklarasi tipe data adalah:

```
type pengenal = tipe;
```

dengan *pengenal* : nama pengenal yang menyatakan tipe data.

tipe : tipe data yang berlaku dalam Turbo Pascal.

Tipe data
- Tipe sederhana
  - Tipe ordinal
  - Tipe integer
  - Tipe boolean
  - Tipe char
  - Tipe subrange
  - Tipe terbilang
  - Tipe real
- Tipe string
- Tipe terstruktur
  - Tipe larik
  - Tipe rekaman
  - Tipe objek
  - Tipe himpunan
  - Tipe berkas
- Tipe pointer
- Tipe procedural
- Tipe objek

Gambar 2. 6. 2. Tipe data dalam Turbo Pascal 7.0.

2.6.3. Variabel dan Konstanta

Variabel mewakili suatu nilai data tertentu yang dioperasikan dalam program. Setiap variabel harus dinyatakan tipe datanya.

Bentuk umum deklarasi variabel adalah:

```
var pengenal : tipe_data;
```

dengan *pengenal* : nama variabel yang dideklarasikan.

tipe_data : tipe data yang digunakan.

Deklarasi konstanta menunjukkan nilai yang tetap dari suatu pengenal dan berlaku pada blok dimana deklarasi tersebut dinyatakan.

Bentuk umum deklarasi konstanta adalah:

```
const pengenal = nilai;
```

dengan *pengenal* : nama konstanta.

nilai : nilai konstanta.

2.6.4. Ungkapan

Ungkapan disusun dari sejumlah operator dan operand. Kebanyakan operator Pascal bersifat biner, yang artinya bahwa setiap operator pasti digunakan oleh dua buah operand. Namun ada juga operator yang bersifat tunggal (*unary*), yaitu yang hanya memerlukan sebuah operand. Operator tunggal selalu ditulis sebelum operand, misalnya –*B*.

Ungkapan yang rumit perlu ditentukan urutan operasinya, sebagaimana tersaji dalam Tabel 2.6.1.

Tabel 2. 6. 1. Urutan operasi.

Operator	Urutan	Kategori
@, not	Pertama (tertinggi)	Operator tunggal
*, /, div, mod, and, shl, shr	Kedua	Operator pengali
+, -, or, xor	Ketiga	Operator penjumlah
=, <>, <, >, >=, <=, in	Keempat (terendah)	Operator relasi

Ada tiga aturan mengenai urutan operasi:

- Operand yang terletak di antara dua buah operator yang mempunyai urutan berbeda dikerjakan terlebih dahulu sesuai dengan operator yang mempunyai urutan yang lebih tinggi.
- Operand yang terletak di antara dua buah operator yang mempunyai urutan sama dikerjakan menurut operator yang ditulis di sebelah kirinya.
- Ungkapan yang terletak di dalam tanda kurung dikerjakan terlebih dahulu sebelum dianggap sebagai sebuah operand.

2.6.5. Statemen

Statemen merupakan satuan terkecil suatu program. Statemen terbagi menjadi dua kelompok, yaitu statemen sederhana dan statemen terstruktur.

Statemen sederhana adalah statemen yang tidak berisi statemen yang lain; terdiri dari statemen pemberian (*assignment statement*), statemen prosedur dan statemen `goto`.

Statemen terstruktur adalah statemen yang tersusun dari sejumlah statemen lain yang dieksekusi secara berurutan, secara terkendali atau secara berulang.

2.6.6. Prosedur dan Fungsi

Prosedur dan fungsi digunakan untuk menambahkan sekelompok statemen yang seolah-olah terpisah dari program utama, tetapi sesungguhnya merupakan bagian dari program utama. Prosedur diaktifkan menggunakan statemen prosedur bersama parameter aktual dan fungsi diaktifkan dengan suatu ungkapan yang hasilnya dikembalikan lagi sebagai nilai baru dari ungkapan tersebut.

Prosedur memiliki struktur yang sama dengan struktur program, dan di dalam prosedur juga dimungkinkan ada prosedur lain yang strukturnya sama. Bentuk ini disebut prosedur tersarang (*nested procedure*). Semua deklarasi dalam prosedur merupakan deklarasi lokal.

Bentuk umum deklarasi prosedur adalah:

`procedure` *nama <(dafpar)>*;

dengan *nama* : nama prosedur.

dafpar : daftar parameter formal.

Deklarasi prosedur dipecah menjadi dua bagian, yaitu nama prosedur dan parameter formal. Parameter formal ada dua macam, yaitu parameter nilai dan parameter variabel. Parameter nilai adalah parameter yang tidak diawali dengan kata baku `var` dan parameter variabel adalah parameter yang diawali dengan `var`.

Sifat parameter nilai adalah bahwa meskipun nilai parameter dalam prosedur mengalami perubahan, tetapi perubahannya tidak dibawa ke dalam program utama. Sedangkan sifat parameter variabel yaitu jika nilai parameter dalam prosedur mengalami perubahan, maka perubahan itu dibawa ke dalam program utama.

Secara umum fungsi hampir sama dengan prosedur, dengan sedikit perbedaan bahwa nama fungsi sekaligus berfungsi sebagai suatu variabel, sehingga dalam deklarasi fungsi harus dinyatakan tipe datanya.

Bentuk umum deklarasi fungsi adalah:

`function` *nama <(dafpar)> : tipe*;

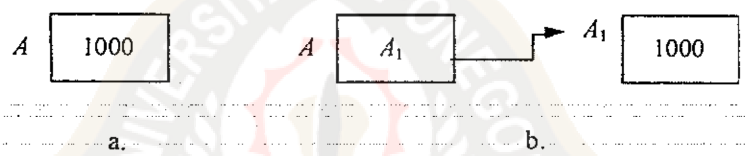
dengan *nama* : nama fungsi.

dafpar : daftar parameter formal.

tipe : tipe data dari fungsi tersebut.

2.6.7. Pointer

Pointer adalah tipe data yang dapat digunakan untuk mengalokasikan dan mendealokasikan memori secara dinamis, yaitu sesuai dengan kebutuhan pada saat program dieksekusi. Variabel dengan tipe pointer disebut juga variabel dinamis. Perbedaan antara variabel dinamis dan variabel statis diilustrasikan pada Gambar 2.6.3.



Gambar 2. 6. 3. Ilustrasi variabel statis dan dinamis.

Pada Gambar 2.6.3.a. variabel A adalah variabel statis. Dalam hal ini 1000 adalah nilai data yang sesungguhnya dan disimpan pada variabel A. Pada Gambar 2.6.3.b. variabel A adalah variabel dinamis. Nilai A_1 pada variabel ini bukan nilai data, melainkan lokasi dimana data yang sesungguhnya berada. Jadi dalam hal ini nilai data yang sesungguhnya tersimpan pada lokasi A_1.

Bentuk umum deklarasi pointer adalah:

```
type pengenal = ^simpul;
     simpul = tipe;
```

dengan *pengenal* : nama pengenal yang menyatakan data bertipe pointer

simpul : nama simpul.

tipe : tipe data dari simpul

2.6.8. Rekursif

Definisi 2. 6. 1. Suatu objek dikatakan ***rekursif*** apabila sebagian objek itu berisi atau didefinisikan sebagai dirinya sendiri. ♦

Algoritma rekursif bisa berjalan apabila persoalan yang diselesaikan, atau nilai fungsi yang dihitung, atau struktur data yang diproses sudah dinyatakan dalam bentuk rekursif. Secara umum, program rekursif dapat dinyatakan sebagai komposisi $\mathcal{P}$ dari statemen dasar S_i (yang tidak berisi P) dan P itu sendiri.

$$P \equiv \mathcal{P}[\,S_i, P\,] \qquad (2.\,6.\,1)$$

Piranti yang perlu dan cukup untuk menyatakan program rekursif adalah prosedur. Jika prosedur P berisi acuan eksplisit ke dirinya sendiri, maka disebut dengan rekursif langsung, sedangkan jika P berisi acuan ke prosedur lain Q, maka disebut sebagai rekursif tak langsung. Dalam hal ini sifat rekursif menjadi tidak begitu terlihat.

Dalam prosedur rekursif juga dimungkinkan adanya komputasi yang tak berhingga, sehingga diperlukan kondisi untuk menghentikan proses eksekusi program. Persyaratan dasarnya jelas bahwa pemanggilan rekursif ke prosedur P harus memperhatikan B yang pada suatu ketika harus bernilai salah. Dengan demikian, algoritma rekursif dapat juga dinyatakan dengan

$$P \equiv \texttt{if}\ B\ \texttt{then}\ \mathcal{P}[\,S_i, P\,] \qquad (2.\,6.\,2)$$

atau

$$P \equiv \mathcal{P}[\,S_i, \texttt{if}\ B\ \texttt{then}\ P\,] \qquad (2.\,6.\,3)$$

2.6.9. Turbo Pascal Grafik

Unit standar Graph menyediakan lebih dari 50 buah rutin grafik yang dapat dipergunakan untuk keperluan pembuatan grafik. Untuk membuat grafik dengan fasilitas tersebut, maka unit standar Graph harus disebutkan dalam program.

Untuk mulai menggunakan grafik, maka prosedur standar `InitGraph` harus disebutkan terlebih dahulu dengan sintak:

```
InitGraph(var GraphDriver : integer);
Var GraphMode  : integer;
    Driverpath : string;
```

`GraphDriver` merupakan driver yang dipergunakan pada komputer. Beberapa konstanta mengenai driver grafik telah didefinisikan di unit standar Graph. Misalnya jika digunakan Color Graphics Adapter (CGA), maka graphics driver yang harus disebutkan adalah CGA atau nilai 1. Kalau pengguna tidak mengetahui adapter grafik yang dipergunakan, dapat dilakukan pendeteksian secara otomatis oleh Turbo Pascal, yaitu dengan menggunakan konstanta `Detect` atau nilai 0.

Untuk berpindah dari keadaan mode grafik ke mode teks dapat dilakukan dengan prosedur standar `RestoreCrtMode`, dan untuk kembali ke keadaan mode grafik lagi dapat dilakukan dengan prosedur standar `SetGraphMode`. Sedangkan untuk mengakhiri penggunaan grafik dan menyebabkan keadaan kembali pada mode layar semula sebelum mode grafik dipergunakan prosedur standar `CloseGraph`.