BAB II

MATERI PENUNJANG

2.1 Koefisien Binomial

Teorema 2. 1. 1

Setiap bilangan n dan k bulat dengan $1 \leq k \leq n-1$ akan memenuhi kesamaan berikut ini

$$\begin{bmatrix} n \\ k \end{bmatrix} = \begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix} \qquad (2.1)$$

Bukti

$$\begin{aligned}
\begin{bmatrix} n \\ k \end{bmatrix} &= \frac{n!}{k!\,(n-k)!} \\
&= \frac{n\,(n-1)!}{k\,(k-1)!\,(n-k)\,(n-k-1)!} \\
&= \frac{(n-k)!}{(k-1)!\,(n-k-1)!} \bullet \frac{n}{k\,(n-k)} \\
&= \frac{(n-1)!}{(k-1)!\,(n-k-1)!} \begin{bmatrix} \frac{1}{k} + \frac{1}{n-k} \end{bmatrix} \\
&= \frac{(n-1)!}{k\,(k-1)!\,(n-k-1)!} + \frac{(n-1)!}{(k-1)!\,(n-k)\,(n-k-1)!} \\
&= \begin{bmatrix} n-1 \\ k \end{bmatrix} + \begin{bmatrix} n-1 \\ k-1 \end{bmatrix}
\end{aligned}$$

terbukti.

■

Bentuk $\begin{bmatrix} n \\ k \end{bmatrix}$ disebut koefisien binomial karena koefisien-koefisien ini memenuhi teorema berikut :

Teorema 2. 1. 2

Misalkan n suatu bilangan bulat positif maka

$$(x+1)^n = 1 + \binom{n}{1}x + \dots + \binom{n}{n-1}x^{n-1} + x^n \qquad (2.2)$$
$$= \sum_{k=0}^{n} \binom{n}{k} x^k$$

Bukti

Dengan langkah induksi, untuk basis induksi n = 1 persamaan (2.2) menjadi

$$(1+x)^1 = \sum_{k=0}^{1} \binom{1}{k} x^k$$
$$= \binom{1}{0} x^0 + \binom{1}{1} x^1$$
$$= 1 + x$$

jadi persamaan (2 . 2) benar untuk nilai n = 1. Misalkan persamaan (2.2) benar untuk nilai n, maka harus dibuktikan persamaan tersebut benar untuk n + 1.

$$(1+x)^{n+1} = (1+x)\left[\sum_{k=0}^{n} \binom{n}{k} x^k\right]$$
$$= \sum_{k=0}^{n} \binom{n}{k} x^k + x \sum_{k=0}^{n} \binom{n}{k} x^k$$

$$= \binom{n}{0}x^0 + \sum_{k=1}^{n}\binom{n}{k}x^k + \sum_{k=o}^{n}\binom{n}{k}x^{k+1}$$

$$= 1 + \sum_{k=1}^{n}\binom{n}{k}x^k + \sum_{k=o}^{n-1}\binom{n}{k}x^{k+1} + \binom{n}{n}x^{n+1} \qquad (2.3)$$

Dengan subtitusi indeks $k + 1 = m$, $k = m-1$ untuk $k = 0$ maka $m = 1$ selanjutnya untuk $k = n-1$ maka $m = n$.

Sehingga $\sum_{k=0}^{n-1}\binom{n}{k}x^{k+1} = \sum_{m=1}^{n}\binom{n}{m-1}x^m$ atau $\sum_{k=1}^{n}\binom{n}{k-1}x^k$.

Sehingga dengan mengganti $\sum_{k=0}^{n-1}\binom{n}{k}x^{k+1}$ dengan $\sum_{k=1}^{n}\binom{n}{k-1}x^k$

Persamaan (2.3) menjadi :

$$(1+x)^{n+1} = 1 + \sum_{k=1}^{n}\binom{n}{k}x^k + \sum_{k=1}^{n}\binom{n}{k-1}x^k + x^{n+1}$$

$$= 1 + \sum_{k=1}^{n}\left[\binom{n}{k} + \binom{n}{k-1}\right]x^k + x^{n+1}$$

$$= 1 + \sum_{k=1}^{n}\binom{n+1}{k}x^k + x^{n+1}$$

$$= \sum_{k=0}^{n+1}\binom{n+1}{k}x^k$$

jadi persamaan (2.2) benar untuk nilai $n + 1$. Sehingga berdasarkan langkah induksi teorema 2.1.2. terbukti.

■

Contoh

Buktikan bahwa :

$$1\binom{n}{1} + 2\binom{n}{2} + \dots + n\binom{n}{n} = n2^{n-1} \qquad (2.5)$$

Penyelesaian :

Untuk membuktikan ini dapat dilakukan dengan megekspansikan bentuk berikut :

$$(1 + x)^n = \binom{n}{0} + \binom{n}{1}x + \binom{n}{2}x^2 + \ldots + \binom{n}{n}x^n \qquad (2.6)$$

Kemudian dilakukan proses diferensiasi terhadap peubah x dari kedua ruas persamaan (2.6), maka diperoleh hasil :

$$n(1 + x)^{n-1} = \binom{n}{1} + 2\binom{n}{2}x + \ldots + n\binom{n}{n}x^{n-1} \qquad (2.7)$$

Dengan mengganti nilai $x = 1$, maka persamaan (2.7) terbukti.

Teorema 2. 1. 3

Jika n bilangan bulat positif. Maka untuk setiap nilai x dan y akan dipenuhi $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$

Bukti

$$(x + y)^n = \left[x\left(1 + \frac{y}{x}\right)\right]^n$$
$$= x^n \left(1 + \left(\frac{y}{x}\right)\right)^n$$

Dengan teorema 2.1.2. diperoleh

$$x^n \sum_{k=0}^{n} \binom{n}{k} \left(\frac{y}{x}\right)^k = x^n \sum_{k=0}^{n} \binom{n}{k} y^k x^{-k}$$
$$= \sum_{k-0}^{n} \binom{n}{k} x^{n-k} y^k$$

Terbukti. ■

2. 2. Relasi Rekursif dan Fungsi Pembangkit

Relasi rekursif yang tergantung nilai n adalah adanya barisan bilangan a_0, a_1, a_2 dan seterusnya sampai a_n, dimana nilai a_n dapat dinyatakan sebagai fungsi dari n nilai a_i sebelumnya.

Contoh :

Barisan Fibonacci : 1, 1, 2, 3, 5, 8, …

Dapat diperoleh dengan menggunakan nilai rekursif : $a_r = a_{r-1} + a_{r-2}$ dengan syarat awal $a_0 = 1$ dan $a_1 = 1$

Definisi 2. 2. 1

Relasi rekursif linier homogen tingkat k dengan koefisien konstan adalah relasi rekursif dalam bentuk

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k} \qquad (2.6)$$

untuk $k \in$ N. Dengan $c_1, c_2, \ldots, c_k$ bilangan real dan $c_k \neq 0$.

Teorema 2. 2. 1

a_1, a_2, c_1, c_2, $\in$ R ,jika $\{p_n\}$ dan $\{q_n\}$ merupakan penyelesaian relasi rekursif

$a_n = c_1 a_{n-1} + c_2 a_{n-2}$ dan $b_1, b_2 \in$ R, maka $s_n = b_1 p_n + b_2 q_n$ untuk setiap $n \in N$ merupakan penyelesaian relasi rekursif.

Bukti

Misalkan $\{p_n\}$ dan $\{q_n\}$ penyelesaian relasi rekursif

$a_n = c_1 a_{n-1} + c_2 a_{n-2}$ maka

$$p_n = c_1 p_{n-1} + c_2 p_{n-2} \qquad (2.7)$$

$$q_n = c_1 q_{n-1} + c_2 q_{n-2} \qquad (2.8)$$

dengan mengalikan persamaan (2. 7) dengan b_1 dan persamaan (2. 8) dengan b_2 diperoleh :

$b_1p_n = b_1c_1p_{n-1} + b_1c_2p_{n-2}$ (2. 9)

$b_2q_n = b_2c_1q_{n-1} + b_2c_2q_{n-2}$ (2.10)

sehingga :

$$\begin{aligned} s_n &= b_1p_n + b_2q_n \\ &= b_1c_1p_{n-1} + b_1c_2p_{n-2} + b_2c_1q_{n-1} + b_2c_2q_{n-2} \\ &= c_1(b_1p_{n-1} + b_2q_{n-1}) + c_2(b_1p_{n-2} + b_2q_{n-2}) \\ &= c_1\, s_{n-1} + c_2s_{n-2} \end{aligned}$$

Jadi s_n merupakan penyelesaian relasi rekursif.

Definisi 2. 2. 2

Persamaan $r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_{k-1}r - c_k = 0$, untuk $i = 1, 2, \ldots, k$ dan $a_i, c_i \in$ R disebut persamaan karakteristik relasi rekursif linier homogen $a_n = c_1a_{n-1} + c_2a_{n-2} + \ldots + c_ka_{n-k}$.

Teorema 2. 2. 2

Untuk setiap $i = 1, 2, \ldots, k$ $a_i, c_i \in$ R. Bilangan $r_1 \neq 0$ merupakan akar persamaan karakteristik

$r^k - c_1r^{k-1} - c_2r^{k-2} - \ldots - c_{k-1}r - c_k = 0$

pada $a_n = c_1\, a_{n-1} + c_2\, a_{n-2} + \ldots + c_k\, a_{n-k}$ jika dan hanya jika barisan $\{p_n\}$ didefinisikan $p_n = (r_1)^n$ adalah penyelesaian dari relasi rekursif.

Bukti

($\rightarrow$) Misalkan bahwa r_1 adalah akar persamaan karakteristik

$$r^k - c_1 r^{k-1} - c_2 r^{k-2} - \dots c_{k-1} r - c_k = 0 \qquad (2.11)$$

pada relasi rekursif $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$ (2.12)

dengan mengambil ruas kanan persamaan (2. 12) maka diperoleh :

$$c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} = c_1 r^{n-1} + c_2 r^{n-2} + \dots + c_k r^{n-k}$$

$$= c_1 (r_1)^{n-1} + c_2 (r_1)^{n-2} + \dots + c_k (r_1)^{n-k}$$

$$= (r_1)^{n-k} \left[c_1 (r_1)^{k-1} + c_2 (r_1)^{k-2} + \dots + ck \right]$$

$$= (r_1)^{n-k} \left[(r_1)^k \right]$$

$$= (r_1)^n$$

sehingga $\{p_n\}$ yang didefinisikan $p_n = (r_1)^n$ merupakan penyelesaian relasi rekursif.

($\leftarrow$) Misalkan barisan $\{p_n\}$ didefinisikan dengan $p_n = (r_1)^n$ merupakan penyelesaian rekursif dari $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$ maka persamaan karakteristiknya :

$$(r_1)^n = c_1 (r_1)^{n-1} + c_2 (r_1)^{n-2} + \dots + c_k (r_1)^{n-k}$$

$$(r_1)^n - c_1 (r_1)^{n-1} - c_2 (r_1)^{n-2} - \dots - c_k (r_1)^{n-k} = 0$$

$$(r_1)^{n-k} \left[(r_1)^k - c_1 (r_1)^{k-1} - c_2 (r_1)^{k-1} - c_k \right] = 0$$

kedua sisi dibagi dengan r_1^{n-k}, sehingga diperoleh :

$$(r_1)^k - c_1 (r_1)^{k-1} - c_2 (r_1)^{k-2} - \dots - c_k = 0$$

jadi r_1 adalah akar persamaan karakteristik .

terbukti. ■

Definisi 2.2.3

Misalkan a_0, a_1, a_2, … a_n, adalah barisan dengan indeks n, maka fungsi pembangkit atau *generating function* dari barisan tersebut didefinisikan sebagai berikut :

$$F(x) = a_0 + a_1x + a_2x + ... + a_nx^n \qquad (2.12)$$

$$= \sum_{n=0}^{\infty} a_n x^n$$

Teorema 2.2. 3

Jika G_1 adalah fungsi pembangkit untuk barisan a_0, a_1, a_2, … dan G_2 fungsi pembangkit untuk barisan b_0, b_1, b_2, … maka $c_1G_1 + c_2G_2$ adalah fungsi pembangkit untuk $c_1a_0 + c_2b_0, c_1a_1 + c_2b_1, \ldots$

Bukti

Ambil G_1 fungsi pembangkit untuk barisan a_0, a_1, a_2, … dan G_2 fungsi pembangkit untuk barisan b_0, b_1, b_2, … maka

$G_1(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n + \ldots$ dan

$G_2(x) = b_0 + b_1x + b_2x^2 + \ldots + b_nx^n + \ldots$

Sehingga

$(c_1G_1 + c_2G_2)(x) = (c_1a_0 + c_2b_0) + (c_1a_1 + c_2b_1)x + \ldots + (c_1a_n + c_2b_n)x^n + \ldots$

terbukti. □

Kebanyakan relasi rekursif yang melibatkan a_n dapat diubah ke dalam suatu persamaan (2.12). Sehingga dapat diselesaikan secara aljabar dari ekspresi yang diperoleh untuk F(x) dengan diekspansikan ke dalam deret pangkat untuk memperoleh a_n yang merupakan koefisien dari x^n. Beberapa persamaan yang berhubungan dengan ekspansi polinomial adalah

$$\frac{1 - x^{n+1}}{1 - x} = 1 + x + x^2 + x^3 + \dots + x^n$$

$$\frac{1}{1 - x} = 1 + x + x^2 + x^3 + \dots + x^n + \dots$$

$$(1 + x)^n = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \binom{n}{3}x^3 + \dots + \binom{n}{r}x^r + \dots + \binom{n}{n}x^n$$

$$\frac{1}{(1 - x)^n} = 1 + \binom{1 + n - 1}{1}x + \binom{2 + n - 1}{2}x^2 + \dots + \binom{r + n - 1}{r}x^r + \dots$$

Contoh

Tentukan koefisien dari x^{16} pada ekspansi

$$(x^2 + x^3 + x^4 + \dots)^5 \tag{2.13}$$

Penyelesaian

Untuk menyelesaikan persoalan ini dikeluarkan suku x^2 dari polinomial menjadi

$$(x^2 + x^3 + x^4 + \dots)^5 = (x^2(1 + x + x^2 + \dots))^5$$

$$= \left(\frac{x^2}{1 - x}\right)^5$$

$$= x^{10}\,\frac{1}{(1 - x)^5} \tag{2.14}$$

Koefisien dari x^{16} pada fungsi pembangkit (2. 13) akan sama dengan koefisien x^6 dari bentuk $(1-x)^{-5}$, yaitu $\binom{6+5-1}{6}$. Jadi koefisien dari x^{16} adalah $\binom{6+5-1}{6}$.

Contoh

Tentukan fungsi pembangkit g(x) dengan koefisien-koefisienya memenuhi hubungan rekursif $a_n = a_{n-1} + a_{n-2}$ dengan syarat awal $a_1 = 1$ dan $a_2 = 2$.

Penyelesaian

Dengan menggunakan hubungan rekursif yang diberikan maka dapat dihitung syarat awal baru yaitu $a_0 = 1$ dan $a_1 = 1$. Kemudian dengan menggunakan jumlahan dari deret pangkat akan diperoleh :

$$g(x) - a_0 - a_1 x = \sum_{n=2}^{\infty} a_n x^n$$

$$= \sum_{n=2}^{\infty} \left[a_{n-1} x^n + a_{n-2} x^n \right]$$

$$= x \sum_{n=2}^{\infty} a_{n-1} x^{n-1} + x^2 \sum_{n=2}^{\infty} a_{n-2} x^{n-2}$$

$$= x \sum_{m=1}^{\infty} a_m x^m + x^2 \sum_{k=0}^{\infty} a_k x^k$$

$$= x \left[g(x) - a_0 \right] + x^2 g(x)$$

Dengan mengganti nilai $a_0 = 1$ dan $a_1 = 1$. Maka diperoleh persamaan fungsi berikut :

$$g(x) - 1 - x = x(g(x) - 1) + x^2 g(x)$$

sehingga : $g(x) = \frac{1}{1 - x - x^2}$

2. 3 Konsep Dasar pohon

2. 3. 1 Graph

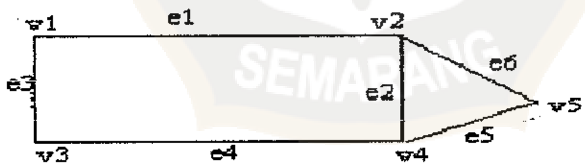
Definisi 2. 3. 1. 1

Suatu graph G yang dinotasikan dengan G = (V, E) adalah himpunan titik-titik (verteks) V, V = $\{v_1, v_2, v_3, \dots, v_n\}$ yang berhingga dan tidak kosong dan himpunan garis yang anggota-anggotanya himpunan bagian dari V x V.

Definisi 2. 3. 1. 2

Graph berarah *(directed graph)* adalah graph yang semua garis-garisnya mempunyai arah. Sebaliknya disebut graph tak berarah atau graph saja.

Contoh



Gambar 2.3.1 Sebuah graph

Gambar 2. 3. 1 menunjukkan sebuah graph dengan

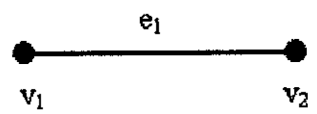
$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{ e_1, e_2, e_3, e_4, e_5, e_6\}$

Definisi 2. 3. 1. 3

Titik-titik v_i dan v_j disebut titik akhir *(endpoint)* dari e_r jika e_r menghubungkan titik v_i dan v_j.

Contoh



Gambar 2. 3. 2

Dari gambar 2. 3. 2 v_1 dan v_2 merupakan titik akhir dari e_1.

Definisi 2. 3.1. 3

Suatu garis e_r disebut incident dengan v_j jika v_j merupakan titik akhir dari e_r.

Contoh

Dari gambar 2. 3. 1 garis e_1 dan e_2 incident dengan titik v_2.

Definisi 2. 3. 1. 4

Dua titik v_j dan v_i pada V disebut adjacent jika dihubungkan oleh sebuah garis.

Contoh

Pada gambar 2. 3. 1 titik v_1 adjacent dengan v_2.

Definisi 2. 3. 1. 5

Dua garis disebut adjacent jika mereka incident pada titik yang sama.

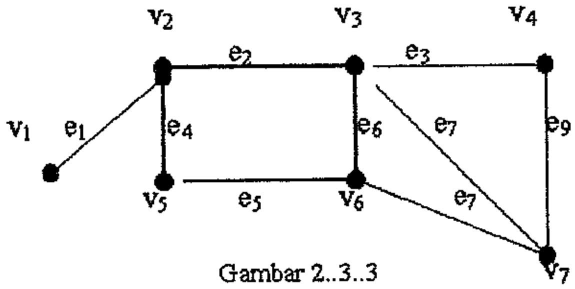
Contoh

Pada gambar 2. 3. 1 e_1 dan e_2 adjacent.

Definisi 2. 3. 1. 6

Derajad(degree) v_i , jika $v_i \in$ V yang dinotasikan deg (v_i) adalah banyaknya edge (garis) yang incident dengan titik v_i.

Contoh



Gambar 2..3..3

Dari Gambar 2. 3. 3 diperoleh :

$\text{Deg}(v_2) = \deg(v_6) = \deg(v_7) = 3$

$\text{Deg}(v_1) = 1$

Definisi 2. 3. 1. 7

Graph G = (V, E) adalah graph lengkap *(complete)* asalkan bahwa untuk setiap pasangan v_i dan v_j anggota V adjacent. Jika G =(V, E) adalah graph lengkap dan banyaknya anggota V adalah n , maka G disebut graph lengkap atas n titik.

Definisi 2. 3. 1. 6

Walk adalah deretan bergantian simpul dan edge yang dimulai dan diakhiri dengan simpul.

Trail adalah walk yang edge-edgenya berbeda.

Path adalah trail yang titik-titiknya berbeda.

Contoh

Dari gambar 2. 3. 3 diperoleh :

Walk yang panjangnya 4 : $v_1e_1v_2e_2v_3e_3v_4e_9v_7$ merupakan trail.

Path $v_1e_1v_2e_4v_5e_5v_6e_7v_7e_9v_4$.

Definisi 2. 3. 1. 9

Graph G adalah terhubung (*connected*) jika $v_i = v_j$ atau jika terdapat sebuah path yang menghubungkan v_i dan v_j dalam G.

Definisi 2. 3. 1. 10

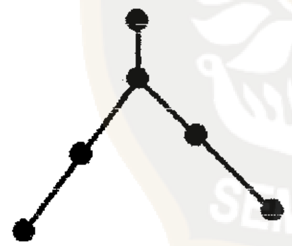
Sirkuit (sikel) adalah graph terhubung yang setiap titiknya (simpul) berderajad dua. Untuk pembahasan selanjutnya istilah simpul (titik) diganti dengan simpul.

2. 3. 2 Pohon (*tree*)

Definisi 2. 3. 2. 1

Pohon (*tree*) adalah suatu graph terhubung (connected) yang tidak memuat sirkuit (sikel).

Contoh



Gambar 2. 3. 4

Gambar 2. 3. 4 menunjukkan pohon yang mempunyai enam simpul.

Teorema 2. 3. 2. 1

Jika G = (V, E) adalah sebuah tree dan u_i dan v_j anggota-anggota V yang berlainan, maka terdapat path tunggal (*unique*) antara titik u_i dan v_j.

Bukti

G = (V, E) adalah sebuah tree, maka setiap simpulnya terhubung dan tidak memuat sirkuit. Akan dibuktikan bahwa terdapat sebuah path tunggal antara u_i dan v_j, jika $u_i, v_j \in V$ dengan kontradiksi.

Adaikan terdapat dua path P_1 dan P_2 yang berlainan menghubungkan u_i dan v_j dalam G, sehingga $P_1 \neq P_2$. Ambil sebuah edge e_1 yang berada dalam path P_1 sedemikian sehingga e_1 tidak berada dalam path P_2. Maka $(P_1 \cup P_2) - \{e_1\}$ terhubung. Akibatnya $(P_1 \cup P_2) \cup \{e_1\}$ adalah suatu sirkuit. Pengandaian salah, yang benar G sebuah tree maka G path antara u_i dan v_j adalah tunggal.

Teorema 2. 3. 2. 2

Jika G = (V, E) adalah sebuah tree . Jika setiap pasangan simpul u dan v yang berlainan pada G terdapat sebuah path tunggal yang menghubungkan u dan v, maka G terhubung dan cacah edgenya (ε) = banyaknya simpul ($\vee$) – 1.

Bukti

Untuk setiap simpul u dan v pada G yang berlainan terdapat sebuah path tunggal yang menghubungkan simpul u dan v maka jelas bahwa G terhubung. Tinggal membuktikan bahwa cacah edge (ε) = cacah simpul ($\vee$) – 1.

Pembuktian dilakukan dengan langkah induksi. Ambil

S = { n$\in$ N : jika graph G dengan n simpul sedemikian sehingga untuk setiap pasangan simpul u dan v pada G yang berlainan terdapat sebuah path tunggal yang menghubungkan u dan v, maka cacah garis (ε) = cacah simpul ($\vee$) – 1}. Untuk basis induksi $\vee = 1$,jelas bahwa G tidak memiliki edge. Untuk basis induksi $\vee = n$ maka banyaknya edge adalah n-1. Selanjutnya harus dibuktikan untuk $\vee = n+1$ berlaku bahwa $\varepsilon = \vee - 1$.

Ambil G = (V, E) graph dengan cacah simpul n+1. Jika u dan v anggota-anggota V yang berlainan maka terdapat sebuah path tunggal yang menghubungkan u dan v. Misalkan sebuah edge e = {a, b} $\in$ E, apabila aeb adalah path tunggal yang menghubungkan a dan b, maka dalam subgraph G – {e} tidak memuat path yang menghubungkan a dan b, G – {e} tidak terhubung. Ambil dua buah komponen dalam subgraph G – {e}, misalkan G_1 dan G_2 komponen pada subgraph G – {e}. Maka untuk setiap i = 1, 2 dan setiap pasangan simpul c, d pada G_i terdapat path tunggal yang menghubungkan c dan d dalam G_i (apabila terdapat dua path yang menghubungkan c dan d dalam G_i , maka terdapat dua path yang menghubungkan c dan d dalam G). Karena untuk setiap i = 1, 2 cacah simpul pada G_i kurang dari atau sama dengan n dan {1, 2, 3, … ,n} $\subseteq$ S, sehingga cacah edge dalam $G_i = \nu(G_i) - 1$. Sedangkan cacah edge dalam G adalah

$$\begin{aligned} \varepsilon(G) &= \varepsilon(G - \{e\}) + 1 \\ &= \varepsilon(G_1) + \varepsilon(G_2) + 1 \\ &= \nu(G_1) - 1 + \nu(G_2) - 1 + 1 \\ &= \nu(G) - 1 \end{aligned}$$

Sehingga untuk cacah simpul n+1 berlaku pula $\varepsilon = \nu - 1$.

Terbukti.

Teorema 2. 3. 2. 3

Jika G = (V, E) graph terhubung yang tidak memuat sikel dengan n simpul sedemikian sehingga $\varepsilon = \nu - 1$, maka G = (V, E) adalah sebuah tree.

Bukti

Bukti dengan kontradiksi. G graph terhubung sedemikian sehingga $(\varepsilon) = v - 1$. Andaikan G memuat sebuah sikel C_1, ambil e_1 sebagai garis dalam C_1 sehingga $G - \{e_1\}$ terhubung dan banyaknya simpul dalam $G - \{e_1\}$ sama dengan banyaknya simpul dalam G, tetapi banyaknya edge $(G - \{e_1\})$ = banyaknya edge dalam (G) –1. Selanjutnya apabila $G - \{e_1\}$ memuat sebuah sikel C_2 dan e_2 pada C_2 sehingga $G - \{e_1, e_2\}$ terhubung, dan banyaknya simpul $G - \{e_1, e_2\}$ = banyaknya simpul (G), tetapi cacah edge $(G - \{e_1, e_2\})$ = cacah edge (G) –1. Sehingga kalau proses dilanjutkan terus akan diperoleh graph terhubung H dengan n vertek yang tidak memuat sikel tetapi cacah garisnya kurang dari n –1. Pengandaian salah , yang benar graph terhubung tanpa sikel adalah tree dengan cacah garis sama dengan cacah simpul dikurangi satu.

Definisi 2. 3. 2. 2

Jika uv adalah edge berarah dalam suatu pohon berakar, maka u disebut simpul ayah dari v, dan v simpul anak dari u. Simpul dalam pada pohon berakar adalah simpul yang mempunnyai anak.

Definisi 2. 3. 2. 3

Jika u dan v adalah simpul- simpul pohon berakar, maka v adalah keturunan u untuk $v \neq u$ dan u adalah simpul pada path tunggal dari akar r ke v.

Definisi 2. 3. 2. 4

Apabila u adalah sebuah simpul pada pohon berakar r, subpohon dengan akar u adalah pohon yang memuat u, cabang u,semua edge berarah dari u ke cabang-cabang u dan semua edge berarah dari cabang u ke cabang-cabangnya.

Definisi 2.3. 2. 5

Jika uv adalah edge berarah dalam sebuah graph berarah, maka u disebut simpul awal dan v adalah simpul akhir.

Definisi 2.3. 2. 6

Apabila v adalah simpul pada graph berarah G, indegree v adalah cacah edge berarah pada G yang menuju v dan outdegree v adalah cacah edge berarah pada G yang keluar dari v.

Definisi 2. 3. 2. 7

Sebuah pohon biner T yang mempunyai dua buah anak, denagn anak pertama disebut subpohon kiri dan anak kedua disebut subpohon kanan.

2. 4 Notasi big O

Diberikan S, himpunan semua fungsi yang mempunyai domain D, dengan D = N, Z atau R. Fungsi f(x) dan g(x) anggota dari S, dikatakan bahwa f(x) = O (g(x)) dibaca " f(x) adalah big O dari g(x)", jika dapat ditemukan bilangan bulat positif c dan k sedemikian sehingga

$$|f(x)| \leq c\,|g(x)|$$

untuk semua $x \in D$ dan $x > k$.

Suatu fungsi f(x) termasuk dalam himpunan O(g(x)) apabila dapat ditemuakan bilangan positif c dan k sedemikian sehingga f(x) kurang dari c. g(x) untuk suatu harga x>k. Penulisan f(x) = O (g(x)) menunjukkan bahwa f(x) anggota dari O (g(x)) atau f(x) ∈ g(x).

Contoh

Diberikan S himpunan semua fungsi yang bernilai riildengan domain R, fungsi f(x) dan g(x) anggota dari S dan didefinisikan dengan

$f(x) = x^3 + 7x^2 + 11x$ dan $g(x) = x^3$ maka $f(x) = O(g(x))$

Penyelesaian:

Ambil c = 18 maka untuk x > 1

$$\begin{aligned} |f(x)| &= x^3 + 7x^2 + 11x \\ &\le x^3 + 7x^3 + 11x^3 \\ &= 18x^3 \\ &\le c.\ g(x) \end{aligned}$$

Dengan demikian $f(x) = O(g(x))$.

Teorema 2. 4.1

Diberikan suatu fungsi polinomial berderajad n,

$f(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$ dan

$g(x) = x^n$, maka $f(x) = O(g(x))$.

Bukt

Dipilih $C = |a_n| + |a_{n-1}| + |a_{n-2}| + \ldots + |a_1| + |a_0|$

$x>1, x \in R$.

$$\begin{aligned} |f(x)| &= |a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0| \\ &\le |an|\ x^n + |a_{n-1}|x^{n-1} + |a_{n-2}|\ x^{n-2} + \ldots + |a_1|\ x + |a_0| \\ &\le |a_n|x^n + |a_{n-1}|x^n + |a_{n-2}|x^n + \ldots + |a_1|x^n + |a_0|x^n \\ &\le (|a_n| + |a_{n-1}| + |a_{n-2}| + \ldots + |a_1| + |a_0|)\ x^n \\ &= c.\ x^n \\ &= c\ .\ g(x) \end{aligned}$$

terbukti $f(x) = O(g(x))$

$\square$

Teorema 2. 4. 2

Diberikan S, himpunan semua fungsi dengan domain D, D = N, Z, atauR dengan kodomain $(0, \infty)$. Fungsi-fungsi $f_1(x)$, $f_2(x)$, $g_1(x)$ dan $g_2(x)$ anggota – anggota S, sedemikian sehingga $f_1(x) = O(g_1(x))$ dan $f_2(x) = O(g_2(x)$ maka berlaku :

a. $.f_1(x) + f_2(x) = O(\max\{ g_1(x), g_2(x)\})$

b. $f_1(x) . f_2(x) = O(g_1(x). g_2(x))$

Bukti

a. Karena $f_1(x) = O(g_1(x))$ dan $f_2(x) = O(g_2(x)$, maka terdapat c_1 dan c_2, k_1 dan k_2 sedemikian sehingga jika $x \in D$ dan $x > k_1$, maka berlaku :

$$\begin{aligned} |f_1(x)| + |f_2(x)| &= |f_1(x)| + |f_2(x)| \\ &\leq c_1 |g_1(x)| + c_2 |g_2(x)| \\ &= c_1 . g_1(x) + c_2 . g_2(x) \\ &\leq c_1 . \max\{ g_1(x), g_2(x)\} + . c_2 . \max\{ g_1(x), g_2(x)\} \\ &= (c_1 + c_2) \max\{ g_1(x), g_2(x)\} \\ &= c. \max\{ g_1(x), g_2(x)\} \end{aligned}$$

maka terbukti bahwa :

$$f_1(x) + f_2(x) = O(\max\{ g_1(x), g_2(x)\})$$

b. Karena $f_1(x) = O(g_1(x))$ dan $f_2(x) = O(g_2(x))$, maka terdapat c_1, c_2, k_1 dan k_2 bilangan bulat positif sedemikian sehingga jika $x \in D$ dan $x > k_1$

maka $|f_1(x)| \leq c_1 . g_1(x)$ dan jika $x > k_2$ berlaku $|f_2(x)| \leq c_2 . g_2(x)$.

Selanjutnya dipilih $k = \max(k_1, k_2)$ dan $c = c_1 . c_2$ maka :

$$
\begin{aligned}
|(f_1 . f_2)(x)| &= |f_1(x)| . |f_2(x)| \\
&\leq c_1 . |g_1(x)| . c_2 . |g_2(x)| \\
&= c_1 . c_2 \; |(g_1 . g_2)(x)| \\
&= c . |g_1(x) . g_2(x)|
\end{aligned}
$$

sehingga :

$$f_1(x) . f_2(x) = O((g_1 . g_2)(x))$$

terbukti

2.5 Algoritma

Algoritma didefinisikan sebagai himpunan berhingga instruksi- instruksi yang apabila ditelusuri, akan menghasilakan suatu penyelesaian dari tugas khusus. Oleh karena itu suatu algoritma harus memenuhi kriteria berikut :

1. Finitenes

   Apabila instruksi- instruksi suatu algoritma di telusuri, maka untuk semua kasus , algoritma akan berhenti setelah diberikan sejumlah berhingga langkah-langkah.

2. Effektiveness

   Setiap instruksi dalam suatu algoritma harus dapat dilaksanakan (executable) langsung mengarah kesasaran yang dituju.

3. Definiteness

   Setiap instruksi dalam suatu algoritma harus jelas dan tidak mempunyai arti ganda (unambiogus).

4. Adanya input

   Suatu algoritma harus mempunyai kualitas nol atau lebih yang disukai secara eksternal.

5. Adanya output

   Suatu algoritma harus menghasilkan minimal satu hasil.

2.6 Menghitung Running time

Untuk menyelesaikan suatu permasalahan, salah satu faktor yang menjadi pertimbangan adalah kecepatan algoritma yang dipilih dalam menyelesaikan permasalahan yang dihadapi. dalam suatu analisa algoritma digunakan istilah running time, yang dinotasikan dengan T(n). Untuk menentukan running time dari suatu algoritma tergantung dari beberapa faktor antara lain :

1. Masukan atau input yang digunakan .
2. Kemampuan kompiler yang digunakan dalam menjalankan program.
3. Kecepatan dari mesin dalam menjalankan instruksi dari suatu program.
4. Kompleksitas waktu dari algoritma yang digunakan.

Running time dari suatu algoritma tergantung terhadap masukkan atu input yang akan diselesaikan, sehingga dikatakan bahwa running time suatu algoritma didefinisikan sebagai suatu fungsi masukkan, masukkan yang dimaksud adalah jumlah data yang akan diproses. Untuk faktor kedua dan ketiga, bahwa running time tergantung dari kompiler dan mesin yang digunakan dalam menyelesaikan program yang merupakan faktor yang mempengaruhi kecepatan proses. Akan tetapi kedua faktor tersebut bersifat sangat teknis dan tergantung pada kompiler yang digunakan. Oleh karena itu, dalam menentukan running time suatu algoritma diasumsikan bahwa runing time yang diperoleh merupakan jumlahan operasi yang dilaksanakan oleh komputer yang ideal.

Salah satu cara untuk menentukan jumlah opersi yang dilaksanakan adalah dengan menggunakan notasi big O atau O(f(n)). Di mana penggunaan parameter n merupakan karakteristik dari n masukkan yang diberikan pada algoritma yang umumnya merupakan jumlah data, maka penulisan O(f(n)) menunjukkan jumlah operasi yang dilakukan untuk menyelesaikan algoritma. Dalam hal ini kompleksitas waktu algoritma tersebut adalah f(n).

Dalam melakukan analisa algoritam terdapat tiga hal yang sering diperhitungkan yaitu :

1. Best case runnning time, yaitu suatu keadaan di mana algoritma diselesaiakn dalam waktu singkat atau baik.
2. Worst case running time, yaitu suatu keadaan terburuk di mana algoritma memerlukan waktu paling lama dalam menyelesaikan proses.
3. Average case running time, yaitu analisa waktu rata-rata yang digunakan untuk melaksanakan suatu algoritma. Ini jarang dilakukan, karena adanya data yang secara umum dapat mewakili keadaan rata-rata data yang digunakan.

Selanjutnya dalam menentukan running time suatu algoritma dipergunakan worst case running time. Hal ini berdasarkan beberapa alasan yaitu :

1. worst case running time merupakan waktu maksimal yang diperlukan suatu algoritma dalam menyesaiakn suatu masalah proses.

2. Worst case running time dari suatu algoritma merupakan suatu batas atas dari suatu running time. Dengan demikian akan memberikan suatu jaminan bahwa algoritma yang digunakan akan mempunyai akhir proses.
3. Untuk beberapa algoritma worst case running time merupakan kejadian yang sering terjadi. Sebagai contoh, dalam proses pencarian database untuk mendapatkan suatu informasi, worst case running time selalu terjadi ketika informasi yang dibutuhkan tak berada dalam database.

Salah satu alat yang digunakan untuk mengetahui effisiensi sebuah algoritma adalah waktu yang cigunakan oleh komputer untuk menyelesaikan sebuah permasalahan dengan menggunakan algoritma tersebut, ketika sebuah nilai input ukurannya telah dispesifikasikan.

Pernyataan tentang hal tersebut diatas termasuk dalam kompleksitas perhitungan sebuah algoritma. sebuah analisa memori yang diinginkan ternmasuk dalam kompleksitas ruang pada algoritma. Pertimbangan kompleksitas waktu dan ruang pada sebuah algoritma adalah penting ketika sebuah algoritma akan diimplementasikan . Jelas sekali ketika suatu algoritma akan menghasilakan jawaban dalam microsecond, menit atau jutaan tahun. Demikian juga memori yang di inginkan harus tersedia untuk menyesaikan suatu permasalahan sehingga kompleksitas ruang dapat digunakan.