

1. PENDAHULUAN

LOGIKA MATEMATIKA, HIMPUNAN DAN FUNGSI

Matematika Diskrit (*discrete mathematics*) adalah salah satu cabang dari matematika yang mempelajari objek-objek diskrit. Diskrit artinya tidak terhubung (*unconnected*), misalnya bilangan riil adalah objek terhubung sedangkan bilangan bulat adalah objek yang tidak terhubung. Diantara contoh permasalahan yang ditangani oleh Matematika Diskrit dapat dilihat pada pertanyaan-pertanyaan berikut ini:

- Ada berapa buah kombinasi *password* yang mungkin untuk mengakses sebuah komputer?
- Berapakah peluang seseorang untuk memenangkan undian?
- Apakah dua komputer dalam suatu jaringan saling terhubung?
- Yang manakah lintasan terpendek antara dua kota yang dihubungkan dengan jalan raya?
- Bagaimana cara mengurutkan sekumpulan bilangan bulat?
- Ada berapa banyak langkah yang diperlukan untuk mengurutkan bilangan-bilangan tersebut?
- Bagaimana cara merancang suatu rangkaian digital?
- Ada berapa banyak alamat internet (*IP address*) yang bisa dibuat?

Mengapa Matematika Diskrit perlu dipelajari? Saat ini komputer digital telah menjadi bagian penting dari kehidupan manusia. Komputer beroperasi secara diskrit dengan unit terkecil yg disebut sebagai bit. Dengan demikian, baik rangkaian penyusun komputer dan juga operasi eksekusi (algoritma) hanya dapat dijelaskan dengan Matematika Diskrit. Untuk mempelajari Matematika Diskrit diperlukan beberapa perangkat atau dasar matematika yang perlu dikuasai, yaitu:

- Logika Matematika (*Mathematical Logic*)
- Teori Himpunan (*Set Theory*)
- Fungsi (*Functions*)
- Barisan (*Sequences*)

yang semuanya pernah dipelajari di tingkat dasar. Bagian pendahuluan ini meninjau ulang perangkat matematika tersebut dan menambahkan beberapa konsep yang diperlukan untuk pelajaran selanjutnya..

LOGIKA MATEMATIKA

Proposisi

Didalam matematika, yang dimaksud dengan logika adalah suatu sistem matematika yang didasarkan pada proposisi. Logika berguna untuk melakukan penalaran matematika (*mathematical reasoning*) dan dalam teknik elektro berguna untuk mendesain rangkaian digital. Sistem logika matematika dibangun dari proposisi-proposisi dan operator-operator. Proposisi adalah sebuah pernyataan yang bisa bernilai *benar* (*true/T/1*) atau *salah* (*false/F/0*) tetapi tidak sekaligus keduanya. Dikatakan bahwa nilai logika (*truth value*) dari sebuah proposisi adalah *benar* atau *salah*. Jadi proposisi adalah sebuah pernyataan dan pernyataan adalah suatu kalimat, tetapi tidak sebaliknya, tidak semua kalimat adalah pernyataan dan tidak semua pernyataan adalah proposisi. Konsep ini akan diperjelas oleh contoh-contoh berikut.

Contoh 1.1.1: “*Gajah lebih besar daripada semut*”

Kalimat ini adalah suatu pernyataan yang merupakan suatu proposisi karena memiliki nilai kebenaran (T/F) yang pasti. Sedangkan nilai kebenaran dari pernyataan ini adalah *benar* (atau *true* atau T atau 1).

Contoh 1.1.2: “*520 < 111*”

Kalimat ini adalah suatu pernyataan yang merupakan suatu proposisi karena memiliki nilai kebenaran (T/F) yang pasti. Sedangkan nilai kebenaran dari pernyataan ini adalah *salah* (atau *false* atau F atau 0).

Contoh 1.1.3: “*y > 5*”

Kalimat ini adalah suatu pernyataan akan tetapi bukan sebuah proposisi karena nilai kebenarannya (T/F) masih harus ditentukan oleh nilai variabel *y*. Pernyataan yang demikian disebut sebagai kalimat terbuka atau fungsi proposisi.

Contoh 1.1.4: “*Sekarang tahun 2003 dan $99 < 5$* ”

Kalimat ini adalah suatu pernyataan yang merupakan suatu proposisi karena memiliki nilai kebenaran (T/F) yang pasti. Nilai kebenaran dari pernyataan ini adalah *salah* (atau *false* atau F atau 0).

Contoh 1.1.5: “*Tolong jangan tidur selama kuliah berlangsung*”

Kalimat ini bukanlah suatu pernyataan melainkan permintaan. Dengan sendirinya ini jelas bukan suatu proposisi.

Contoh 1.1.6: “ *$x < y$ jika dan hanya jika $y > x$* ”

Meskipun terdapat variabel di dalamnya, kalimat ini adalah pernyataan yang berupa proposisi karena memiliki nilai kebenaran (T/F) yang pasti, tidak bergantung pada nilai spesifik x maupun y . Nilai kebenaran dari pernyataan ini adalah *benar*.

Operator Logika dan Proposisi Gabungan

Beberapa contoh terdahulu menunjukkan bahwa beberapa proposisi dapat digabungkan menjadi sebuah *proposisi gabungan*. Hal ini kita formalisasikan dengan melambangkan proposisi sebagai p , q , r , atau s , dan memperkenalkan operator-operator logika. Kita akan membahas operator-operator berikut:

- Operator negasi (NOT)
- Operator konjungsi (AND)
- Operator disjungsi (OR)
- Operator eksklusif OR (XOR)
- Operator implikasi (jika – maka)
- Operator bikondisional (jika dan hanya jika)

Tabel logika (*truth table*) dapat dipakai untuk menunjukkan cara operator-operator tersebut menggabungkan beberapa proposisi menjadi sebuah proposisi gabungan.

Operator negasi (NOT)

Operator ini dilambangkan dengan symbol “ \neg ”, adalah sebuah operator uner (*unary*), yaitu operator yang hanya memerlukan satu masukan. Tabel logika dari operator ini adalah

p	$\neg p$
T	F
F	T

Operator konjungsi (AND)

Operator ini diberi lambang “ \wedge ”, adalah suatu operator biner (*binary*), yaitu operator yang membutuhkan dua masukan. Tabel logikanya adalah

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Operator AND hanya bernilai benar jika kedua masukannya bernilai benar.

Operator disjungsi (OR)

Operator ini diberi lambang “ \vee ”, adalah suatu operator biner. Tabel logika dari operator ini adalah

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Operator OR hanya bernilai salah jika kedua proposisinya salah.

Operator OR eksklusif (XOR)

Operator ini diberi lambang “ \oplus ”, adalah suatu operator biner. Tabel logika dari operator ini adalah

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Operator XOR hanya bernilai *benar* jika kedua proposisinya memiliki nilai logika yang berlainan.

Operator implikasi (jika-maka)

Operator ini diberi lambang “ \rightarrow ”, adalah suatu operator biner. Tabel logika dari operator ini adalah

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Perhatikan bahwa operator ini hanya bernilai *salah* jika proposisi pertama (p) bernilai benar tetapi memberikan implikasi yang *salah*. Implikasi tetap bernilai benar bahkan jika kedua proposisi bernilai salah.

Operator bikondisi (jika dan hanya jika)

Operator ini diberi lambang " \leftrightarrow ", adalah suatu operator biner. Tabel logika dari operator ini adalah

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Operator bikondisi bernilai benar jika kedua proposisi bernilai benar atau bernilai salah. Hasil diatas bisa dianalisis dengan prinsip kesetangkupan, yakni melihat jika-maka kearah maju dan ke arah mundur.

Dengan beberapa operator yang diperkenalkan diatas, kita dapat membentuk suatu proposisi gabungan. Nilai logika dari proposisi ini dapat diperiksa dengan tabel logika. Sebagai contoh, dari dua proposisi p dan q , serta operator negasi, konjungsi dan disjungsi, kita dapat membentuk dua proposisi baru berikut $(\neg p) \vee (\neg q)$ dan $\neg(p \wedge q)$. Berikut ini tabel logikanya.

p	q	$\neg p$	$\neg q$	$(\neg p) \vee (\neg q)$	$(p \wedge q)$	$\neg(p \wedge q)$
T	T	F	F	F	T	F
T	F	F	T	T	F	T
F	T	T	F	T	F	T
F	F	T	T	T	F	T

Jika diperhatikan, nilai logika dari $(\neg p) \vee (\neg q)$ dan $\neg(p \wedge q)$ adalah sama. Ini mengindikasikan bahwa kedua proposisi gabungan ini ekuivalen. Namun, secara formal ekivalensi perlu ditunjukkan dengan mengevaluasi kedua proposisi jika digabungkan dengan operator bikondisi \leftrightarrow , seperti diperlihatkan pada tabel berikut ini

p	q	$\neg(p \wedge q)$	$(\neg p) \vee (\neg q)$	$\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$
T	T	F	F	T
T	F	T	T	T
F	T	T	T	T
F	F	T	T	T

Dari tabel diatas, disimpulkan bahwa proposisi $\neg(p \wedge q)$ dan $(\neg p) \vee (\neg q)$ adalah ekivalen secara logis, karena $\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$ selalu bernilai *benar*. Lebih lanjut, hal ini membawa kepada suatu konsep dalam logika matematika yang disebut sebagai *tautologi*. Suatu tautologi adalah pernyataan yang selalu bernilai *benar*. Contoh dari tautologi adalah:

- $r \vee (\neg r)$
- $\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$

Contoh pertama bersifat trivial dan mudah diperiksa, sedangkan contoh kedua sudah ditunjukkan oleh tabel yang telah kita buat sebelumnya. Jika proposisi gabungan $s \rightarrow t$ sebuah tautologi, maka kita bisa menuliskannya sebagai $s \Rightarrow t$ dan jika $s \leftrightarrow t$ sebuah tautologi, maka kita bisa menuliskannya sebagai $s \Leftrightarrow t$.

Lawan dari tautologi adalah *kontradiksi*, yaitu suatu proposisi yang selalu bernilai salah. Contoh dari kontradiksi adalah:

- $r \wedge (\neg r)$
- $\neg(\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q))$

Karena kontradiksi selalu bernilai salah, maka negasinya selalu bernilai benar, demikian pula sebaliknya. Dengan demikian, negasi dari sebarang tautologi adalah sebuah kontradiksi. Dan sebaliknya negasi dari sebuah kontradiksi adalah sebuah tautologi.

Latihan: Kita sudah menunjukkan bahwa $\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$ adalah tautologi. Dengan cara yang sama tunjukkan bahwa $\neg(p \vee q) \leftrightarrow (\neg p) \wedge (\neg q)$ juga suatu tautologi. Kedua tautologi ini disebut sebagai hukum De Morgan !

Fungsi Proposisi

Fungsi proposisi atau kalimat terbuka adalah pernyataan yang mengandung satu buah variabel atau lebih. Nilai logika dari fungsi ini bergantung pada nilai yang kita berikan kepada variabel tersebut. Sebagai contoh, tinjau fungsi proposisi $x - 3 > 5$. Fungsi proposisi ini bisa kita tuliskan sebagai $P(x)$, dimana P adalah predikat dan x adalah variabel. Nilai logika $P(x)$ untuk $x=2$ adalah *salah*, untuk $x=8$ juga bernilai *salah*, sedangkan untuk $x=9$ maka $P(x)$ bernilai *benar*, demikian seterusnya.

Tinjau suatu fungsi proposisi $Q(x, y, z)$ yang didefinisikan sebagai $x + y = z$. Disini, Q adalah predikat, sedangkan $x, y,$ dan z adalah variabelnya. Nilai logika untuk $Q(x=2, y=3, z=5)$ atau bisa kita tulis secara sederhana sebagai $Q(2,3,5)$ adalah *benar*, $Q(0,1,2)$ adalah *salah* dan $Q(9,-9,0)$ bernilai *benar*. Nilai-nilai lain bisa diperiksa dari fungsi proposisi tersebut.

Kuantifikasi Universal (\forall) dan Kuantifikasi Eksistensial (\exists)

Tinjau fungsi proposisi $P(x)$. Pernyataan yg dikuantifikasi secara universal akan berbunyi:

"Untuk semua x dalam semesta pembicaraan, $P(x)$ adalah benar."

Dalam matematika logika dikenal pengkuantifikasi universal dengan lambang " \forall ". Pernyataan tersebut diatas juga bisa dituliskan sebagai:

$\forall x P(x)$ dibaca : "untuk semua $x P(x)$ " atau "untuk setiap $x P(x)$ "

Sebagai catatan, $\forall x P(x)$ bisa benar atau salah, jadi merupakan *sebuah proposisi*, bukan *fungsi proposisi*. Sebagai contoh, perhatikan proses kuantifikasi universal dari proposisi berikut.

$S(x)$: x adalah seorang mahasiswa ITB.

$G(x)$: x adalah orang yang pandai.

Maka, proposisi $\forall x (S(x) \rightarrow G(x))$ dapat dibaca sebagai:

“Jika x mahasiswa ITB, maka x adalah orang yang pandai” atau
“Semua mahasiswa ITB pandai”

Disamping kuantifikasi univierseal yang telah dijelaskan, ada juga kuantifikasi eksistensial. Suatu pernyataan yang di-kuantifikasi secara eksistensial akan berbunyi seperti

”Ada x didalam semesta pembicaraan dimana $P(x)$ benar.”

Dengan peng-kuantifikasi eksistensial \exists , kita bisa menuliskan proposisi tersebut sebagai

$\exists x P(x)$ dibaca “Ada sebuah x sedemikian hingga $P(x)$.” atau
 “Ada sedikitnya sebuah x sedemikian hingga $P(x)$.”

Pernyataan $\exists x P(x)$ bisa bernilai benar atau salah tetapi tidak sekaligus keduanya, jadi pernyataan ini merupakan sebuah proposisi, bukan fungsi proposisi. Sebagai contoh, tinjau pernyataan berikut:

$P(x)$: “ x adalah penduduk Indonesia”
 $G(x)$: “ x adalah orang yang pandai”

Maka, pernyataan $\exists x (P(x) \wedge G(x))$ bisa dibaca sebagai:

“Ada x sedemikian hingga x adalah penduduk Indonesia dan x pandai.” atau
“Sedikitnya satu orang penduduk Indonesia adalah orang yang pandai.”

Berikutnya, kita tinjau contoh dari pernyataan dalam matematika. Misalkan semesta pembicaraan kita adalah bilangan riil. Pernyataan $\forall x \exists y (x + y = 320)$ dapat dibaca sebagai

“Untuk setiap x ada y sehingga $x + y = 320$.”

Pernyataan ini bernilai *benar*. Akan tetapi, jika semesta pembicaraan adalah bilangan cacah, maka pernyataan ini *salah*. Mengapa demikian? (Hint: bilangan cacah adalah $\{0, 1, 2, \dots\}$)

dan tidak ada bilangan negatif). Sehubungan dengan hal ini, kita bisa menunjukkan kesalahan (menyanggah atau *disproof*) dari proposisi tersebut dengan suatu sanggahan (*counter example*)

Sanggahan dari $\forall x P(x)$ adalah sebuah nilai argumen (objek) c sehingga $P(c)$ salah. Pernyataan seperti $\forall x (P(x) \rightarrow Q(x))$ dapat disanggah secara mudah dengan memberikan sanggahannya. Sebagai contoh, pernyataan “Semua burung bisa terbang.” dapat disanggah dengan memberikan sanggahan “Penguin”.

Kedua buah pengkuantifikasi yang telah dijelaskan terhubung melalui negasi berikut:

$\neg(\forall x P(x))$ ekuivalen scr logis dengan $\exists x (\neg P(x))$.

$\neg(\exists x P(x))$ ekuivalen scr logis dengan $\forall x (\neg P(x))$.

TEORI HIMPUNAN

Meskipun secara sekilas nampak sebagai dua topik yang terpisah, sebenarnya logika matematika dan teori himpunan memiliki hubungan yang sangat erat. Bagian ini mengulang konsep himpunan yang telah diterima pada pelajaran tingkat menengah dan memformulasikannya dalam kerangka kerja matematika diskrit serta beberapa konsep tambahan yang berkaitan.

Sebuah himpunan adalah suatu kumpulan objek (yang disebut sebagai “elemen” atau anggota himpunan). Untuk menyatakan keanggotaan a dari himpunan A , kita bisa menuliskan

$a \in A$ dibaca “ a adalah elemen dari himpunan A ” atau
“ a adalah anggota dari himpunan A ”

Sebaliknya, untuk menyatakan a sebagai bukan anggota dari A , kita bisa menuliskan

$a \notin A$ dibaca “ a bukan elemen dari himpunan A ” atau
“ a bukan anggota dari himpunan A ”

Keanggotaan bisa juga dinyatakan dengan menyebutkan anggota-anggota himpunan tersebut, misalnya $A = \{a_1, a_2, \dots, a_n\}$. Sebagai catatan, urutan dari penyebutan elemen tidak berpengaruh, begitu pula, frekuensi kemunculan dari elemen yang sama di dalam himpunan juga tidak berpengaruh.

Himpunan A dan B disebut sama, dituliskan sebagai $A=B$, jika dan hanya jika keduanya memiliki elemen yang tepat sama. Contoh:

- $A = \{9, 2, 7, -3\}$, $B = \{7, 9, -3, 2\}$. Maka $A=B$, meski urutan penyebutannya berbeda
- $A = \{\text{anjing, kucing, kuda}\}$, $B = \{\text{kucing, kuda, tupai, anjing}\}$. Dalam hal ini $A \neq B$, karena himpunan A tidak mengandung “tupai”
- $A = \{\text{anjing, kucing, kuda}\}$, $B = \{\text{kucing, kuda, anjing, anjing}\}$. Maka $A=B$, meskipun “anjing” disebut dua kali pada B.

Selanjutnya kita mengingat himpunan-himpunan “baku” berikut

- Bilangan Cacah (*natural number*) $\mathbf{N} = \{0, 1, 2, 3, \dots\}$
- Bilangan Bulat (*integers*) $\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$
- Bilangan Bulat Positif $\mathbf{Z}_+ = \{1, 2, 3, 4, \dots\}$
- Bilangan Riil (*real numbers*) $\mathbf{R} = \{47.3, -12, \pi, \dots\}$
- Bilangan Rasional $\mathbf{Q} = \{1.5, 2.6, -3.8, 15, \dots\}$

Disamping itu perlu dikenal juga himpunan-himpunan dan penulisan notasi himpunan berikut

- $A = \emptyset$ “himpunan kosong/himpunan tanpa anggota”
- $A = \{z\}$, dalam hal ini $z \in A$, tapi $z \neq \{z\}$
- $A = \{\{b, c\}, \{c, x, d\}\}$, yaitu himpunan yang anggotanya adalah juga himpunan
- $A = \{\{x, y\}\}$, yakni $\{x, y\} \in A$, tapi $\{x, y\} \neq \{\{x, y\}\}$
- $A = \{x \mid P(x)\}$ yaitu “himpunan semua x sedemikian hingga $P(x)$ ”
- $A = \{x \mid x \in \mathbf{N} \wedge x > 7\} = \{8, 9, 10, \dots\}$ “notasi pembentuk himpunan”

Sekarang kita bisa mendefinisikan himpunan bilangan rasional Q:

$$\mathbf{Q} = \{a/b \mid a \in \mathbf{Z} \wedge b \in \mathbf{Z}_+\} \quad \text{atau}$$

$$\mathbf{Q} = \{a/b \mid a \in \mathbf{Z} \wedge b \in \mathbf{Z} \wedge b \neq 0\}$$

Sedangkan untuk bilangan riil, kita hanya bisa menuliskannya sebagai:

$$\mathbf{R} = \{r \mid r \text{ adalah bilangan riil}\}$$

karena belum ada cara lain untuk menyatakannya dengan lebih baik.

Himpunan bagian dituliskan dengan notasi " \subseteq ", sedangkan " \subset " melambangkan himpunan bagian sejati. Contoh:

$$A \subseteq B \quad \text{"A adalah himpunan bagian dari B"}$$

$A \subseteq B$ jika dan hanya jika setiap elemen dari A adalah juga elemen dari B, yang penulisannya bisa diformalkan sebagai:

$$A \subseteq B \Leftrightarrow \forall x (x \in A \rightarrow x \in B)$$

Contoh: $A = \{3, 9\}$, $B = \{5, 9, 1, 3\}$, maka $A \subseteq B$

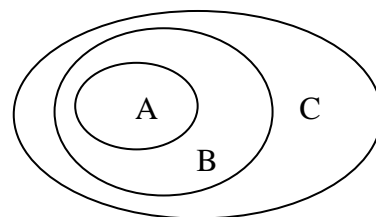
$A = \{3, 3, 3, 9\}$, $B = \{5, 9, 1, 3\}$, maka berlaku juga $A \subseteq B$

$A = \{1, 2, 3\}$, $B = \{2, 3, 4\}$, dan $A \subseteq B$ adalah salah

Untuk himpunan bagian, dikenal aturan-aturan berikut ini:

- $A = B \Leftrightarrow (A \subseteq B) \wedge (B \subseteq A)$
- $(A \subseteq B) \wedge (B \subseteq C) \Rightarrow A \subseteq C$

(lihat Diagram Venn di samping)



Gb1.1 Diagram Venn

- $\emptyset \subseteq A$ untuk sebarang himpunan A
- $A \subseteq A$ untuk sebarang himpunan A

Untuk menyatakan A sebagai himpunan bagian sejati (*proper subset*) dari B, kita menuliskan $A \subset B$ dan dibaca “A adalah himpunan bagian sejati dari B”. Untuk himpunan bagian sejati, berlaku:

$$A \subset B \Leftrightarrow \forall x (x \in A \rightarrow x \in B) \wedge \exists x (x \in B \wedge x \notin A) \quad \text{atau}$$

$$A \subset B \Leftrightarrow \forall x (x \in A \rightarrow x \in B) \wedge \neg \forall x (x \in B \rightarrow x \in A)$$

Besarnya suatu himpunan dinyatakan sebagai kardinalitas (*cardinality*). Jika suatu himpunan memiliki n -buah anggota yang berlainan, $n \in \mathbb{N}$, kita menyebut S sebagai himpunan berhingga dengan kardinalitas n dan dituliskan $|S| = n$. Contoh:

$$A = \{\text{Mercedes, BMW, Porsche}\}, \text{ maka } |A| = 3$$

$$B = \{1, \{2, 3\}, \{4, 5\}, 6\}, \text{ maka } |B| = 4$$

$$C = \emptyset, \text{ maka } |C| = 0$$

$$D = \{x \in \mathbb{N} \mid x \leq 7000\}, \text{ dalam hal ini } |D| = 7001$$

$$E = \{x \in \mathbb{N} \mid x \geq 7000\}, \text{ maka } |E| = \text{tak berhingga}$$

Dalam teori himpunan dikenal pula himpunan kuasa (*power set*). Himpunan kuasa dari A, dituliskan sebagai 2^A atau $P(A)$ adalah suatu himpunan yang mengandung semua himpunan bagian dari A, dengan demikian $2^A = \{B \mid B \subseteq A\}$. Contoh:

$$A = \{x, y, z\}$$

$$2^A = \{\emptyset, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$$

$$A = \emptyset, \text{ maka } 2^A = \{\emptyset\}$$

$$\text{Catatan: Untuk yang terakhir ini } |A| = 0, \text{ dan } |P(A)| = |2^A| = 1$$

Sewaktu diperkenalkan cara penggambaran fungsi dengan grafik, kita mengenal pasangan bilangan. Untuk fungsi y terhadap x , pasangan bilangan tersebut dituliskan sebagai pasangan berurut (x, y) yang disebut juga sebagai tupel (*tuple*). Konsep ini bisa diperumum menjadi n -buah pasangan berurut (*n-ordered tuple*), yaitu $(x_1, x_2, x_3, \dots, x_n)$ yang sebenarnya merupakan hasil perkalian Kartesian n -buah himpunan. Dalam ilmu komputer, pasangan berurut dipakai dalam basis data relasional. Berikut ini formulasi yang lebih formal.

Suatu n -buah pasangan berurut (*ordered n -tuple*) $(a_1, a_2, a_3, \dots, a_n)$ adalah kumpulan berurut dari objek-objek. Dua buah n -tupel berurut $(a_1, a_2, a_3, \dots, a_n)$ dan $(b_1, b_2, b_3, \dots, b_n)$ disebut sama jika dan hanya jika keduanya memiliki elemen-elemen yang tepat sama dalam urutan yang juga sama, yakni, $a_i = b_i$ untuk $1 \leq i \leq n$. Perkalian Kartesian dari dua himpunan didefinisikan sebagai :

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$$

Contoh : $A = \{x, y\}$, $B = \{a, b, c\}$. Maka hasil perkalian Kartesian kedua himpunan ini adalah

$$A \times B = \{(x, a), (x, b), (x, c), (y, a), (y, b), (y, c)\}$$

Selanjutnya perlu diperhatikan pula bahwa:

- $A \times \emptyset = \emptyset$
- $\emptyset \times A = \emptyset$
- Untuk himpunan A dan B yg tidak kosong: $A \neq B \Leftrightarrow A \times B \neq B \times A$
- $|A \times B| = |A| \cdot |B|$

Lebih umum lagi, perkalian Kartesian dari dua himpunan didefinisikan sebagai:

$$A_1 \times A_2 \times \dots \times A_n = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i \text{ untuk } 1 \leq i \leq n\}$$

Dua buah himpunan dapat digabungkan dengan operator \cup . Untuk dua buah himpunan A dan B, penggabungan (*union*) menghasilkan:

$$A \cup B = \{x \mid x \in A \vee x \in B\}$$

Contoh: $A = \{a, b\}$, $B = \{b, c, d\}$ maka $A \cup B = \{a, b, c, d\}$

Demikian pula, dua himpunan A dan B dapat dilakukan operasi interseksi atau irisan \cap , yakni $A \cap B = \{x \mid x \in A \wedge x \in B\}$

Contoh: $A = \{a, b\}$, $B = \{b, c, d\}$ maka $A \cap B = \{b\}$

Dua buah himpunan disebut tak beririsan atau *disjoint* jika irisan dari keduanya adalah himpunan kosong: $A \cap B = \emptyset$.

Perbedaan (pengurangan) antara dua himpunan, A dan B, adalah suatu himpunan yang memiliki elemen-elemen didalam A yang bukan elemen B:

$$A - B = \{x \mid x \in A \wedge x \notin B\}$$

Contoh: $A = \{a, b\}$, $B = \{b, c, d\}$, maka $A - B = \{a\}$

Komplemen dari suatu himpunan A, dituliskan sebagai A' atau \bar{A} , adalah himpunan yang mengandung semua elemen dalam semesta pembicaraan yang tidak ada didalam A:

$$A' = U - A$$

Contoh: $U = \mathbf{N}$, $B = \{250, 251, 252, \dots\}$, maka $B' = \{0, 1, 2, \dots, 248, 249\}$

Bagaimana membuktikan $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$? Ada dua cara yang bisa digunakan yakni dengan pembuktian logika matematik dan dengan menggunakan tabel keanggotaan.

Cara I:

$$x \in A \cup (B \cap C)$$

$$\Leftrightarrow x \in A \vee x \in (B \cap C)$$

$$\Leftrightarrow x \in A \vee (x \in B \wedge x \in C)$$

$$\Leftrightarrow (x \in A \vee x \in B) \wedge (x \in A \vee x \in C)$$

(hukum distributif untuk logika matematika)

$$\Leftrightarrow x \in (A \cup B) \wedge x \in (A \cup C)$$

$$\Leftrightarrow x \in (A \cup B) \cap (A \cup C)$$

Cara II: Menggunakan tabel keanggotaan

1: berarti “ x adalah anggota dari himpunan ini”

0: berarti “ x adalah bukan anggota dari himpunan ini”

A	B	C	$B \cap C$	$A \cup (B \cap C)$	$A \cup B$	$A \cup C$	$(A \cup B) \cap (A \cup C)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Dari contoh tersebut, maka dapat kita simpulkan bahwa setiap ekspresi logis dapat ditransformasikan kedalam ekspresi ekivalen dalam teori himpunan dan begitu pula sebaliknya.

FUNGSI

Suatu fungsi f dari himpunan A ke himpunan B adalah sebuah *assignment* atau pemetaan kepada tepat ke satu elemen dari B dari setiap elemen dari A . Kita tuliskan

$$f(a) = b$$

jika b merupakan elemen unik dari B yang merupakan peta fungsi f dari a anggota A . Jika f adalah fungsi dari A ke B , kita menuliskan

$$f: A \rightarrow B$$

(Catatan, notasi “ \rightarrow ” disini sama sekali tidak berhubungan dengan (operator logika) implikasi jika ... maka).

Pada $f: A \rightarrow B$, kita menyebut A adalah *domain* dari f dan B adalah *codomain* dari f . Jika $f(a)=b$, kita menyebut b sebagai *image* (bayangan) dari a , sedangkan a sendiri adalah *pre-image* dari b . Jangkauan (*Range*) dari pemetaan $f: A \rightarrow B$ adalah himpunan semua bayangan dari elemen A . Dengan demikian, maka di katakan $f: A \rightarrow B$ memetakan A ke B .

Perhatikan suatu fungsi (diskrit) dengan pemetaan $f: P \rightarrow C$ dimana $P=\{Linda, Max, Kathy, Peter\}$ sedangkan $C=\{Boston, New York, Hong Kong, Moscow\}$. Fungsi f didefinisikan sebagai berikut:

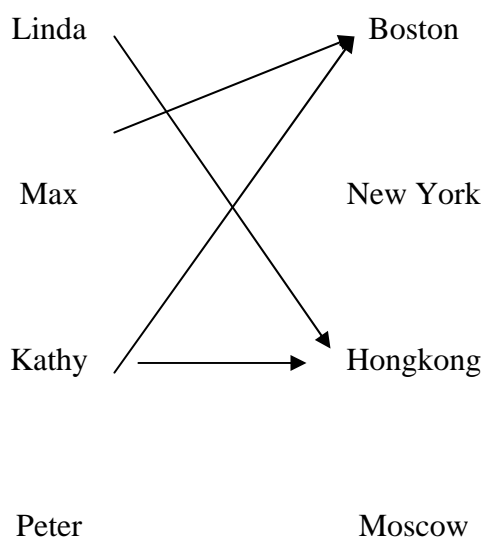
$$\begin{aligned} f(Linda) &= Moscow, & f(Max) &= Boston \\ f(Kathy) &= Hong Kong, & f(Peter) &= New York \end{aligned}$$

Pada kasus ini, maka jangkauan dari f adalah C . Jika sekarang kita punya g dengan pemetaan berikut

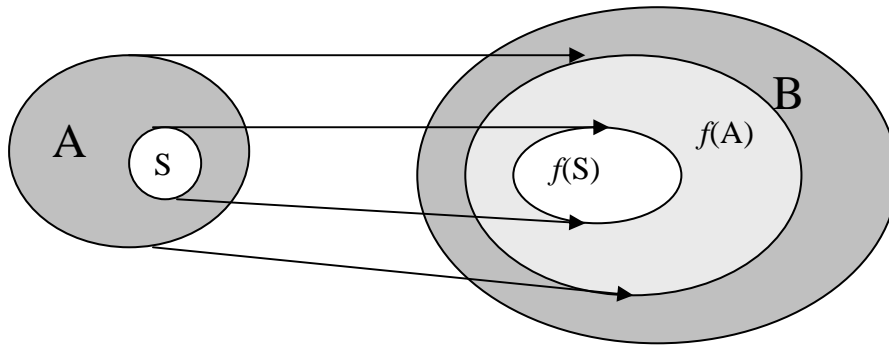
$$\begin{aligned} g(Linda) &= Moscow, & g(Max) &= Boston \\ g(Kathy) &= Hong Kong, & g(Peter) &= Boston, \end{aligned}$$

apakah g masih merupakan fungsi? Perdefinisi g masih merupakan sebuah fungsi, dengan jangkauan $\{Moscow, Boston, Hong Kong\}$. Ada cara lain untuk menyatakan fungsi ini, yaitu dengan tabel dan pemetaan seperti di bawah ini

x	$g(x)$
Linda	Moscow
Max	Boston
Kathy	Hongkong
Peter	Boston



Jika domain dari sebuah fungsi amat besar, maka lebih baik menyatakan f kedalam suatu formula, misalnya $f: \mathbf{R} \rightarrow \mathbf{R}$ dengan $f(x) = 2x$. Evaluasi fungsi ini pada beberapa nilai riil x memberikan hasil, misalnya, $f(1) = 2$, $f(3) = 6$, $f(-3) = -6$, ... dan seterusnya.



Gb. 1.2 Pemetaan himpunan A ke himpunan B. Bayangan dari A adalah $f(A)$ dan bayangan dari $S \subseteq A$ adalah $f(S)$.

Tinjau dua buah fungsi f_1 dan f_2 yang memetakan A ke R. Penjumlahan dan perkalian dari f_1 dengan f_2 adalah juga fungsi dari A ke R yang didefinisikan sebagai berikut

penjumlahan dua fungsi: $(f_1 + f_2)(x) = f_1(x) + f_2(x)$ dan

perkalian dua fungsi: $(f_1 f_2)(x) = f_1(x) f_2(x)$

Sebagai contoh dari operasi ini, tinjau dua fungsi $f_1(x) = 3x$ dan $f_2(x) = x + 5$. Hasil penjumlahan dan perkalian fungsi adalah:

hasil penjumlahan: $(f_1 + f_2)(x) = f_1(x) + f_2(x) = 3x + x + 5 = 4x + 5$

hasil perkalian: $(f_1 f_2)(x) = f_1(x) f_2(x) = 3x(x + 5) = 3x^2 + 15x$

Sudah kita pelajari bahwa jangkauan dari fungsi $f: A \rightarrow B$ adalah himpunan semua bayangan dari elemen $a \in A$. Jika kita hanya meninjau himpunan bagian $S \subseteq A$, maka himpunan semua bayangan dari elemen $s \in S$ disebut sebagai bayangan dari S. Bayangan dari S diberi notasi

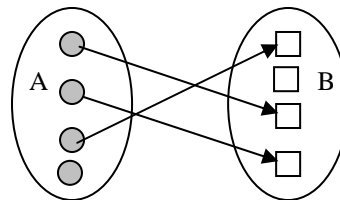
$f(S)$, dengan demikian $f(S) = \{f(s)|s \in S\}$, seperti yang diperlihatkan pada Gambar 1.2. Sebagai contoh, tinjau fungsi f berikut:

$$\begin{aligned} f(\text{Linda}) &= \text{Moscow}, & f(\text{Max}) &= \text{Boston} \\ f(\text{Kathy}) &= \text{Hong Kong}, & f(\text{Peter}) &= \text{Boston} \end{aligned}$$

Untuk $S = \{\text{Linda}, \text{Max}\}$, maka bayangan dari S adalah $f(S) = \{\text{Moscow}, \text{Boston}\}$. Demikian pula untuk $T = \{\text{Max}, \text{Peter}\}$, maka bayangannya adalah $f(T) = \{\text{Boston}\}$.

Suatu fungsi $f: A \rightarrow B$ disebut satu-ke-satu (*injective*), jika dan hanya jika

$$\forall x \in A \text{ dan } y \in B \ (f(x) = f(y) \rightarrow x = y)$$



Gb.1.3 Ilustrasi fungsi injektif

Dengan kata lain, fungsi f adalah satu-ke-satu jika dan hanya jika f tidak memetakan dua elemen berbeda dari A ke dua elemen yang sama dari B . Perhatikan fungsi f dan fungsi g pada contoh berikut ini

$$\begin{aligned} f(\text{Linda}) &= \text{Moscow} \\ f(\text{Max}) &= \text{Boston} \\ f(\text{Kathy}) &= \text{Hong Kong} \\ f(\text{Peter}) &= \text{Boston} \end{aligned}$$

$$\begin{aligned} g(\text{Linda}) &= \text{Moscow} \\ g(\text{Max}) &= \text{Boston} \\ g(\text{Kathy}) &= \text{Hong Kong} \\ g(\text{Peter}) &= \text{New York} \end{aligned}$$

Fungsi f bukan satu-ke-satu karena Max dan Peter dipetakan ke elemen yang sama dari bayangan pemetaan, yaitu Boston.

Fungsi g adalah satu-ke-satu karena setiap elemen domain asal dipetakan ke suatu elemen bayangan yang unik.

Bagaimana cara membuktikan bahwa f satu-ke-satu? Untuk pembuktian ini, terlebih dahulu kita lihat definisinya:

$$\forall x, y \in A (f(x) = f(y) \rightarrow x = y)$$

Contoh: untuk $f: \mathbb{R} \rightarrow \mathbb{R}$, apakah $f(x) = x^2$ satu-ke-satu (*injective*)?

Jawab: Pernyataan ini bisa disanggah dengan memberikan sanggahannya: $f(3) = f(-3)$, tapi $3 \neq -3$, jadi f tidak satu-ke-satu.

Contoh lain: untuk $f: \mathbb{R} \rightarrow \mathbb{R}$, apakah $f(x) = 3x$ injektif?

Jawab: Kita tinjau definisinya. Suatu fungsi $f: A \rightarrow A$ adalah injektif jika berlaku: $\forall x, y \in A (f(x) = f(y) \rightarrow x = y)$. Akan ditunjukkan bahwa $f(x) \neq f(y)$ jika $x \neq y$

$$x \neq y \Leftrightarrow 3x \neq 3y \quad f(x) \neq f(y),$$

Jadi, jika $x \neq y$, maka $f(x) \neq f(y)$, sehingga f adalah fungsi yang injektif.

Suatu fungsi $f: A \rightarrow B$ dengan $A, B \subseteq \mathbb{R}$ disebut *strictly increasing*, jika

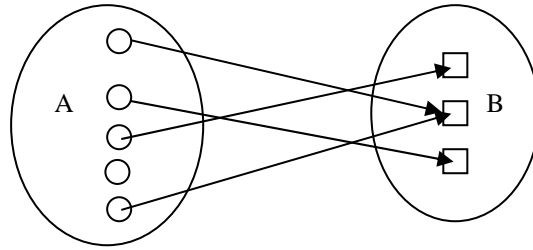
$$\forall x, y \in A (x < y \rightarrow f(x) < f(y)),$$

dan *strictly decreasing*, jika

$$\forall x, y \in A (x < y \rightarrow f(x) > f(y)).$$

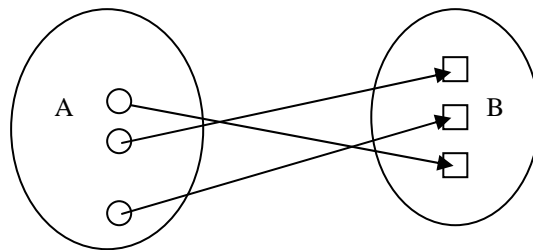
Jelas bahwa suatu fungsi yang *strictly increasing* atau *strictly decreasing* adalah fungsi yang satu-ke-satu.

Suatu fungsi $f: A \rightarrow B$ disebut *onto*, atau surjektif (*surjective*), jika dan hanya jika untuk setiap elemen $b \in B$ ada suatu elemen $a \in A$ dimana $f(a) = b$. Dengan kata lain, f adalah *onto* jika dan hanya jika *range*-nya adalah seluruh bayangan pemetaan.



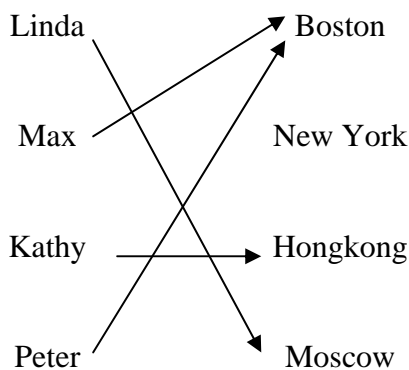
Gb. 1.4 Ilustrasi fungsi yang surjektif (*onto*)

Suatu fungsi $f: A \rightarrow B$ disebut berkorespondensi satu-ke-satu, atau suatu bijeksi (bijection), jika dan hanya jika fungsi tsb sekaligus **satu-ke-satu** dan **onto**.



Gb. 1. 4 Ilustrasi fungsi bijektif (*satu-ke-satu dan onto*)

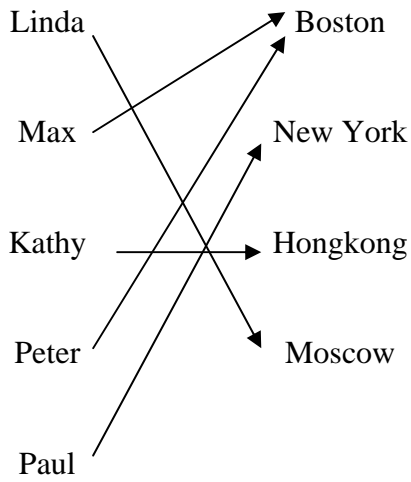
Jelas bahwa, jika f suatu bijeksi, sedangkan A dan B himpunan yang berhingga, maka akan berlaku $|A| = |B|$. Contoh-contoh berikut ini akan memperjelas konsep fungsi yang injektif, onto dan bijektif.



Fungsi f **tidak injektif** (satu-ke-satu) karena Max dan Peter dipetakan ke kota yang sama (Boston).

Fungsi f **tidak surjektif** (onto) karena tidak semua kota mendapatkan pasangan dari orang (New York)

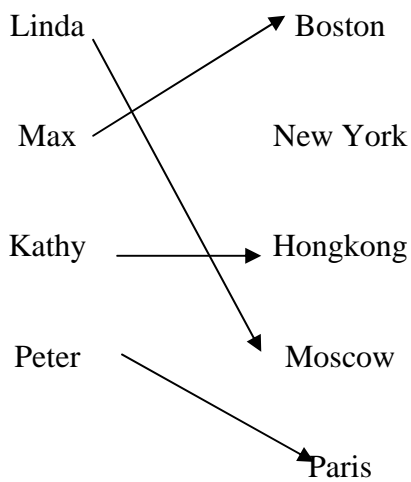
Fungsi f jelas **tidak bijektif** karena baik syarat injektif maupun surjektif tidak terpenuhi.



Fungsi f **tidak injektif** (satu-ke-satu) karena Max dan Peter dipetakan ke kota yang sama (Boston).

Fungsi f **surjektif** (onto) karena semua kota sekarang mendapatkan pasangan dari orang.

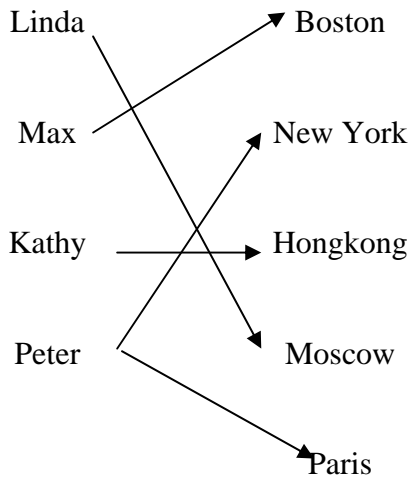
Fungsi f jelas **tidak bijektif** syarat injektif tidak terpenuhi.



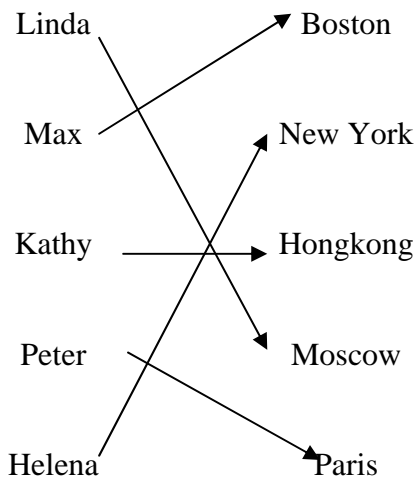
Fungsi f **injektif** (satu-ke-satu) karena orang dan kota telah dipasangkan satu ke satu.

Fungsi f **tidak surjektif** (onto) karena New York tidak dipasangkan ke orang tertentu.

Fungsi f jelas **tidak bijektif** syarat surjektif tidak terpenuhi.



Disini f tidak **injektif**, **tidak surjektif** dan juga **tidak bijektif**. Bahkan f bukanlah suatu fungsi karena Peter dipetakan ke dua kota yang berbeda.

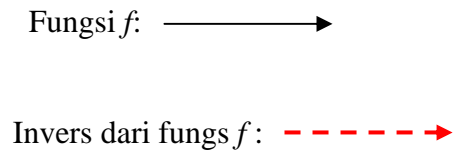
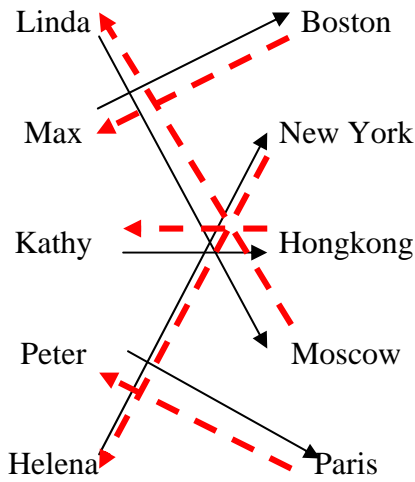


Fungsi f adalah **injektif** karena setiap anggota himpunan orang dipetakan ke tepat satu dari anggota himpunan kota.

Fungsi f adalah juga **surjektif** karena seluruh anggota himpunan bayangan, yaitu himpunan kota, mendapatkan pasangan himpunan orang.

Akhirnya f adalah sebuah fungsi yang **bijektif** karena, baik syarat injektif maupun syarat surjektif telah terpenuhi.

Salah satu sifat penting dari fungsi yang bijektif adalah fungsi tersebut memiliki invers. Fungsi invers dari bijeksi $f: A \rightarrow B$ adalah suatu fungsi $f^{-1}: B \rightarrow A$ dengan $f^{-1}(b) = a$ untuk $f(a) = b$. Invers dari fungsi diilustrasikan pada gambar berikut ini.



Contoh: fungsi f dan inversnya

Fungsi f

- $f(\text{Linda}) = \text{Moscow}$
- $f(\text{Max}) = \text{Boston}$
- $f(\text{Kathy}) = \text{Hong Kong}$
- $f(\text{Peter}) = \text{Lübeck}$
- $f(\text{Helena}) = \text{New York}$

Fungsi f^{-1}

- $f^{-1}(\text{Moscow}) = \text{Linda}$
- $f^{-1}(\text{Boston}) = \text{Max}$
- $f^{-1}(\text{Hong Kong}) = \text{Kathy}$
- $f^{-1}(\text{Lübeck}) = \text{Peter}$
- $f^{-1}(\text{New York}) = \text{Helena}$

Jelas bahwa, f adalah bijektif karena itu memiliki invers.

Inversi hanya mungkin untuk bijeksi. Jadi, fungsi yang bijektif adalah fungsi yang invertibel.

Komposisi dari dua buah fungsi $g:A \rightarrow B$ dan $f:B \rightarrow C$, dinyatakan sebagai $f \circ g$, didefinisikan sebagai $(f \circ g)(a) = f(g(a))$. Ini berarti bahwa fungsi g pertama-tama diterapkan kepada elemen $a \in A$, yang memetakannya ke elemen dari B , kemudian, fungsi f diterapkan pada elemen dari B ini, yang selanjutnya memetakannya ke sebuah elemen dari C . Maka, fungsi komposit tersebut memetakan A ke C .

Contoh: Diberikan dua buah fungsi, yaitu $f(x)=7x-4$, dan $g(x)=3x$, dimana $f: \mathbb{R} \rightarrow \mathbb{R}$, demikian pula $g: \mathbb{R} \rightarrow \mathbb{R}$. Maka komposisi dengan argument $x=5$ akan memberikan

$$(f \circ g)(5) = f(g(5)) = f(15) = 105 - 4 = 101$$

dan untuk sebarang $x \in \mathbb{R}$ akan berlaku $(f \circ g)(x) = f(g(x)) = f(3x) = 21x - 4$.

Komposisi dari suatu fungsi dengan invers-nya dihitung sbb:

$$(f^{-1} \circ f)(x) = f^{-1}(f(x)) = x$$

Dengan demikian, komposisi suatu fungsi dengan invers-nya adalah fungsi identitas $i(x) = x$.

Graf dari suatu fungsi $f: A \rightarrow B$ adalah himpunan pasangan berurut $\{(a, b) | a \in A \text{ and } f(a) = b\}$. Graf tersebut adalah himpunan bagian dari perkalian kartesian A dengan B, yakni $A \times B$, yang dapat dipakai untuk mem-visualisasikan f dalam sistem koordinat dua dimensi.

Fungsi *floor* dan *ceiling* memetakan bilangan riil ke bilangan bulat ($\mathbb{R} \rightarrow \mathbb{Z}$). Fungsi *floor* memetakan suatu bilangan riil $r \in \mathbb{R}$ ke bilangan bulat terbesar $z \in \mathbb{Z}$ dengan $z \leq r$. Notasi dari fungsi *floor* dengan argument r adalah $\lfloor r \rfloor$. Contoh: $\lfloor 2.3 \rfloor = 2$, $\lfloor 2 \rfloor = 2$, $\lfloor 0.5 \rfloor = 0$, $\lfloor -3.5 \rfloor = -4$. Fungsi *ceiling* memetakan suatu bilangan riil $r \in \mathbb{R}$ ke bilangan bulat terkecil $z \in \mathbb{Z}$ dengan $z \geq r$. Notasinya adalah $\lceil r \rceil$. Contohnya $\lceil 2.3 \rceil = 3$, $\lceil 2 \rceil = 2$, $\lceil 0.5 \rceil = 1$, $\lceil -3.5 \rceil = -3$.

BARISAN (SEQUENCES)

Barisan adalah suatu daftar yang terurut (*ordered lists*) dari elemen-elemen. Suatu barisan didefinisikan sebagai fungsi dari himpunan bagian bilangan cacah \mathbb{N} ke suatu himpunan S . Kita memakai notasi a_n untuk melambangkan bayangan dari n dan a_n ini juga disebut sebagai suku dari barisan. Perhatikan contoh berikut

Himpunan bagian dari \mathbb{N} :	1	2	3	4	5	...
	↓	↓	↓	↓	↓	...
S :	2	4	6	8	10	...

Barisan bisa dituliskan pula sebagai $\{a_n\}$. Suku barisan akan lebih jelas lagi jika diberikan sebagai suatu rumus (formula). Misalnya barisan pada contoh di atas dapat dituliskan dengan

lebih spesifik sebagai $\{a_n\}$, dimana $a_n = 2n$. Berikut ini contoh beberapa barisan beserta formulanya

Barisan	Formula
1, 3, 5, 7, 9, ...	$a_n = 2n - 1$
-1, 1, -1, 1, -1, ...	$a_n = (-1)^n$
2, 5, 10, 17, 26, ...	$a_n = n^2 + 1$
0.25, 0.5, 0.75, 1, 1.25 ...	$a_n = 0.25n$
3, 9, 27, 81, 243, ...	$a_n = 3^n$

Barisan yang panjangnya berhingga disebut juga sebagai *string*. Suatu string dinyatakan sebagai $a_1 a_2 a_3 \dots a_n$. Panjang dari sebuah string S adalah banyaknya suku dari string tersebut. Suatu string kosong (*empty string*) sama sekali tidak mengandung suku sehingga panjangnya sama dengan nol.

Suatu barisan bisa dijumlahkan. Notasi $\sum_{j=m}^n a_j$ artinya a_j dijumlahkan dari indeks $j=m$ sampai

dengan $j=n$, jadi $\sum_{j=m}^n a_j = a_m + a_{m+1} + \dots + a_n$. Variabel j disebut sebagai indeks penjumlahan, dan

berjalan dari batas bawah m ke batas atas n . Indeks juga bisa memakai huruf yang lain.

Dengan demikian, penjumlahan 1000 buah suku pertama dari deretan $\{a_n\}$ dng $a_n = n^2$ untuk

$n = 1, 2, 3, \dots$ dapat dituliskan sebagai $\sum_{j=1}^{1000} j^2$. Demikian pula, kita bisa mengevaluasi jumlah

barisan lain, misalnya $\sum_{j=1}^6 j = 1 + 2 + 3 + 4 + 5 + 6 = 21$. Untuk $\sum_{j=1}^{100} j$, Gauss telah menemukan

formulanya, yaitu

$$\sum_{j=1}^n j = \frac{n(n+1)}{2}$$

Dengan demikian, untuk soal diatas diperoleh jawaban: $\sum_{j=1}^{100} j = \frac{100(100+1)}{2} = 5050$.

Hal yang menyangkut proses penjumlahan barisan sering ditemui dalam pemrograman komputer, yaitu suatu lingkaran bersarang (*nested loop*), misalnya dalam penjumlahan *double* atau *triple*. Berikut ini contoh penjumlahan ganda

$$\sum_{i=1}^5 \sum_{j=1}^2 ij = \sum_{i=1}^5 (i + 2i) = \sum_{i=1}^5 3i = 3 + 6 + 9 + 12 + 15 = 45$$