

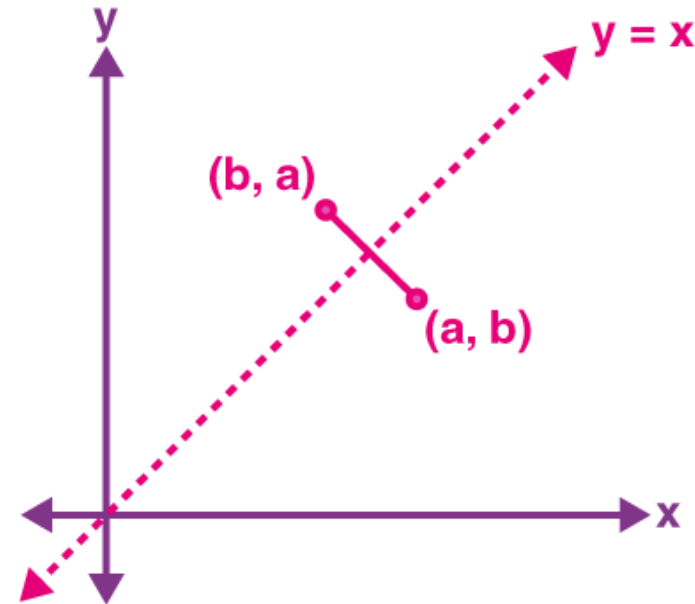
Relasi dan Fungsi

Part 2

Relasi Inversi

- Misalkan R adalah relasi dari himpunan A ke himpunan B . Invers dari relasi R , dilambangkan dengan R^{-1} , adalah relasi dari B ke A yang didefinisikan oleh

$$R^{-1} = \{ (b, a) \mid (a, b) \in R \}$$



Contoh. Misalkan $P = \{2, 3, 4\}$ dan $Q = \{2, 4, 8, 9, 15\}$. Jika kita definisikan relasi R dari P ke Q dengan

$$(p, q) \in R \text{ jika } p \text{ habis membagi } q$$

maka kita peroleh

$$R = \{(2, 2), (2, 4), (4, 4), (2, 8), (4, 8), (3, 9), (3, 15)\}$$

R^{-1} adalah *invers* dari relasi R , yaitu relasi dari Q ke P dengan

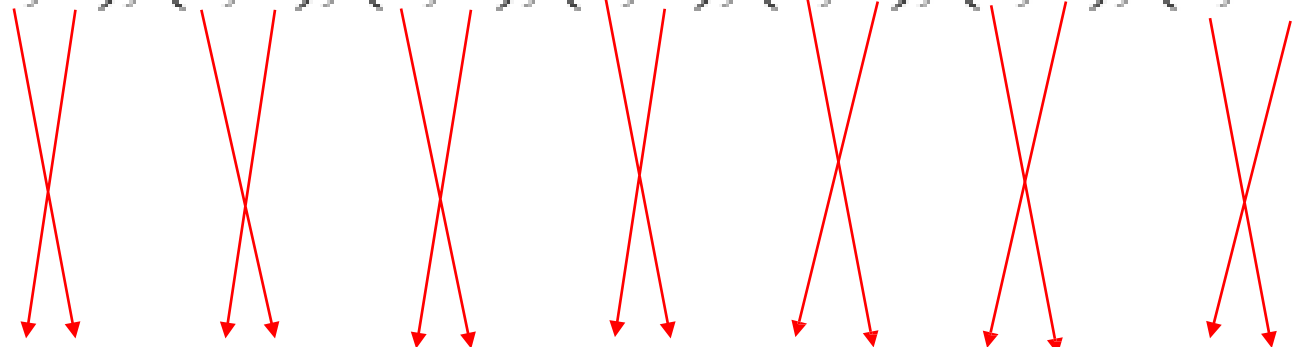
$$(q, p) \in R^{-1} \text{ jika } q \text{ adalah kelipatan dari } p$$

maka kita peroleh

$$R^{-1} = \{(2, 2), (4, 2), (4, 4), (8, 2), (8, 4), (9, 3), (15, 3)\}$$

$$R = \{(2, 2), (2, 4), (4, 4), (2, 8), (4, 8), (3, 9), (3, 15)\}$$

$$R^{-1} = \{(2, 2), (4, 2), (4, 4), (8, 2), (8, 4), (9, 3), (15, 3)\}$$



Jika M adalah matriks yang merepresentasikan relasi R ,

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

maka matriks yang merepresentasikan relasi R^{-1} , misalkan N , diperoleh dengan melakukan *transpose* terhadap matriks M ,

$$N = M^T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Mengkombinasikan Relasi

- Karena relasi biner merupakan himpunan pasangan terurut, maka operasi himpunan seperti irisan, gabungan, selisih, dan beda setangkup antara dua relasi atau lebih juga berlaku.
- Jika R_1 dan R_2 masing-masing adalah relasi dari himpunan A ke himpunan B , maka $R_1 \cap R_2$, $R_1 \cup R_2$, $R_1 - R_2$, dan $R_1 \oplus R_2$ juga adalah relasi dari A ke B .

Contoh 18. Misalkan $A = \{a, b, c\}$ dan $B = \{a, b, c, d\}$.

Relasi $R_1 = \{(a, a), (b, b), (c, c)\}$

Relasi $R_2 = \{(a, a), (a, b), (a, c), (a, d)\}$

$$R_1 \cap R_2 = \{(a, a)\}$$

$$R_1 \cup R_2 = \{(a, a), (b, b), (c, c), (a, b), (a, c), (a, d)\}$$

$$R_1 - R_2 = \{(b, b), (c, c)\}$$

$$R_2 - R_1 = \{(a, b), (a, c), (a, d)\}$$

$$R_1 \oplus R_2 = \{(b, b), (c, c), (a, b), (a, c), (a, d)\}$$

- Jika relasi R_1 dan R_2 masing-masing dinyatakan matriks M_{R_1} dan M_{R_2} , maka matriks yang menyatakan gabungan dan irisan dari kedua relasi tersebut adalah

$$M_{R_1 \cup R_2} = M_{R_1} \vee M_{R_2} \quad \text{dan} \quad M_{R_1 \cap R_2} = M_{R_1} \wedge M_{R_2}$$

Contoh. Misalkan bahwa relasi R_1 dan R_2 pada himpunan A dinyatakan oleh matriks

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \text{ dan } R_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

maka

$$M_{R_1 \cup R_2} = M_{R_1} \vee M_{R_2} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$M_{R_1 \cap R_2} = M_{R_1} \wedge M_{R_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Komposisi Relasi

- Misalkan R adalah relasi dari himpunan A ke himpunan B , dan S adalah relasi dari himpunan B ke himpunan C . Komposisi R dan S , dinotasikan dengan $S \circ R$, adalah relasi dari A ke C yang didefinisikan oleh

$$S \circ R = \{(a, c) \mid a \in A, c \in C, \text{ dan untuk beberapa } b \in B, (a, b) \in R \text{ dan } (b, c) \in S\}$$

Contoh. Misalkan

$$R = \{(1, 2), (1, 6), (2, 4), (3, 4), (3, 6), (3, 8)\}$$

adalah relasi dari himpunan $\{1, 2, 3\}$ ke himpunan $\{2, 4, 6, 8\}$ dan

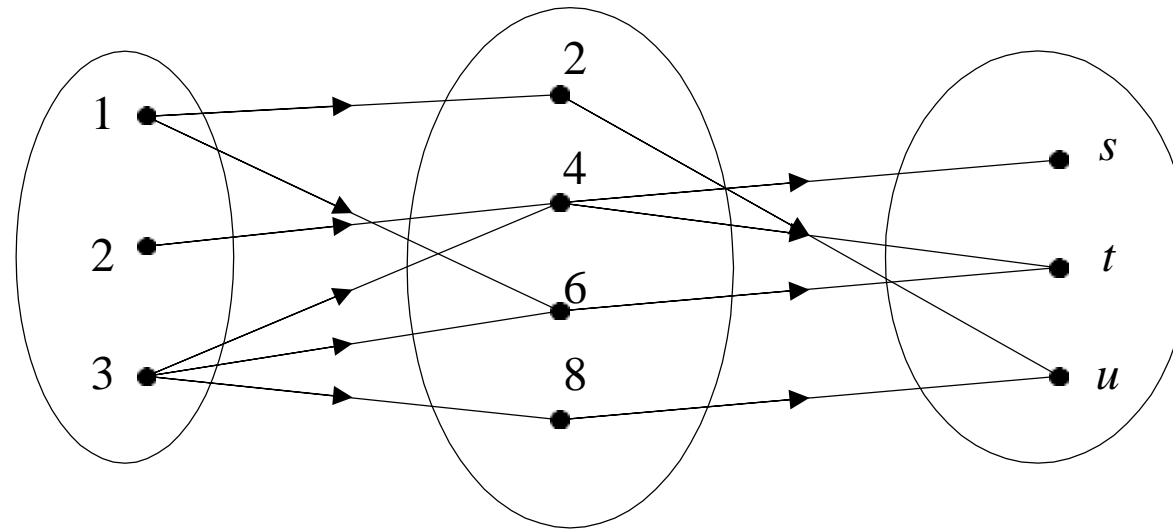
$$S = \{(2, u), (4, s), (4, t), (6, t), (8, u)\}$$

adalah relasi dari himpunan $\{2, 4, 6, 8\}$ ke himpunan $\{s, t, u\}$.

Maka komposisi relasi R dan S adalah

$$S \circ R = \{(1, u), (1, t), (2, s), (2, t), (3, s), (3, t), (3, u)\}$$

Komposisi relasi R dan S lebih jelas jika diperagakan dengan diagram panah:



Relasi n-ary

- Relasi biner hanya menghubungkan antara dua buah himpunan.
- Relasi yang lebih umum menghubungkan lebih dari dua buah himpunan. Relasi tersebut dinamakan relasi *n-ary* (baca: ener).
- Jika $n = 2$, maka relasinya dinamakan relasi biner ($bi = 2$). Relasi *n-ary* mempunyai terapan penting di dalam basisdata.
- Misalkan A_1, A_2, \dots, A_n adalah himpunan. Relasi *n-ary* R pada himpunan-himpunan tersebut adalah himpunan bagian dari $A_1 \times A_2 \times \dots \times A_n$, atau dengan notasi $R \subseteq A_1 \times A_2 \times \dots \times A_n$.
- Himpunan A_1, A_2, \dots, A_n disebut daerah asal relasi dan n disebut **derajat**.

Contoh. Misalkan

$$NIM = \{13598011, 13598014, 13598015, 13598019, \\ 13598021, 13598025\}$$

$$Nama = \{\text{Amir, Santi, Irwan, Ahmad, Cecep, Hamdan}\}$$

$$MatKul = \{\text{Matematika Diskrit, Algoritma, Struktur Data,} \\ \text{Arsitektur Komputer}\}$$

$$Nilai = \{A, B, C, D, E\}$$

Relasi *MHS* terdiri dari 5-tupel (*NIM*, *Nama*, *MatKul*, *Nilai*):

$$MHS \subseteq NIM \times Nama \times MatKul \times Nilai$$

Satu contoh relasi yang bernama *MHS* adalah

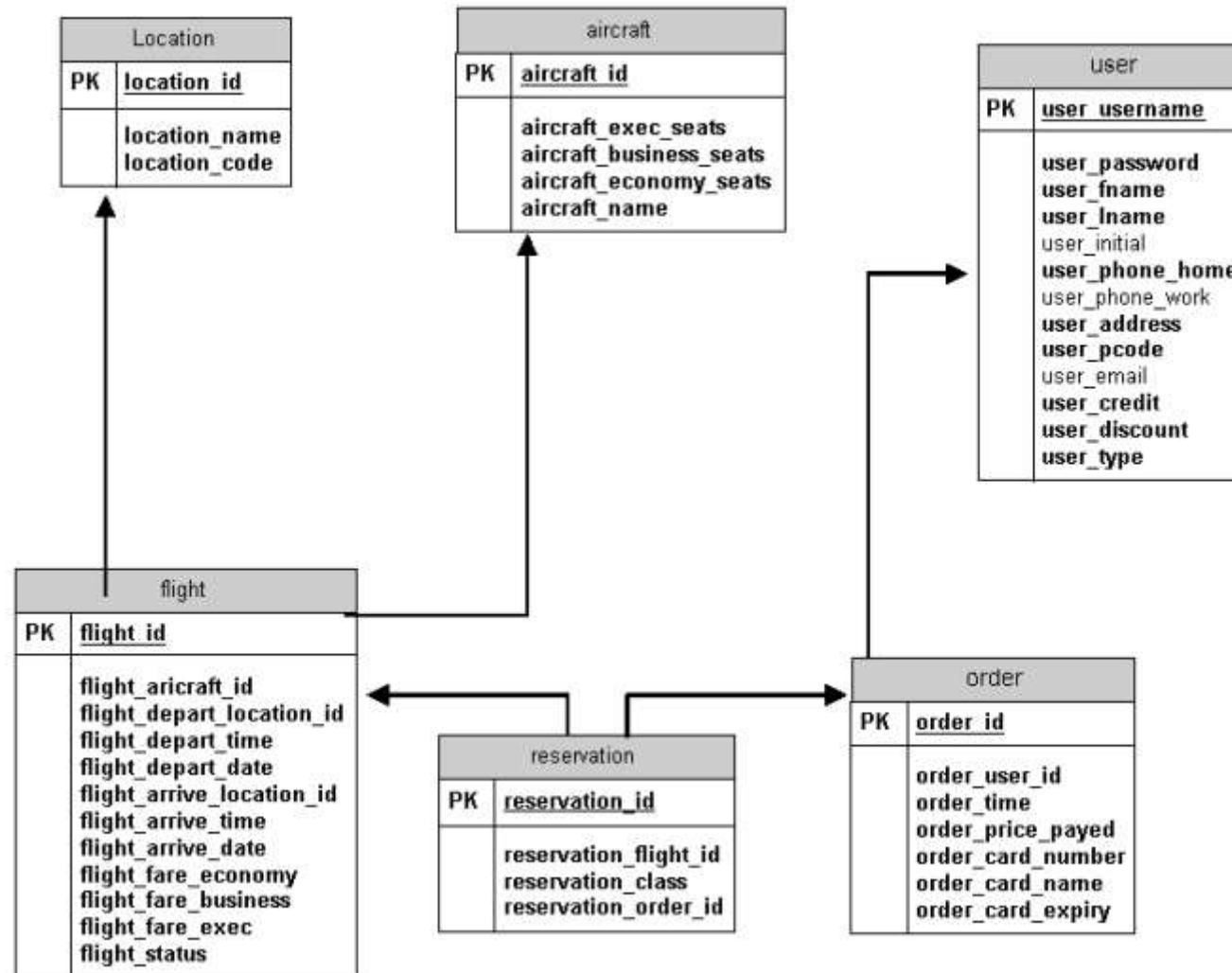
$$\begin{aligned} MHS = \{ & (13598011, \text{Amir}, \text{Matematika Diskrit}, A), \\ & (13598011, \text{Amir}, \text{Arsitektur Komputer}, B), \\ & (13598014, \text{Santi}, \text{Arsitektur Komputer}, D), \\ & (13598015, \text{Irwan}, \text{Algoritma}, C), \\ & (13598015, \text{Irwan}, \text{Struktur Data C}), \\ & (13598015, \text{Irwan}, \text{Arsitektur Komputer}, B), \\ & (13598019, \text{Ahmad}, \text{Algoritma}, E), \\ & (13598021, \text{Cecep}, \text{Algoritma}, A), \\ & (13598021, \text{Cecep}, \text{Arsitektur Komputer}, B), \\ & (13598025, \text{Hamdan}, \text{Matematika Diskrit}, B), \\ & (13598025, \text{Hamdan}, \text{Algoritma}, A, B), \\ & (13598025, \text{Hamdan}, \text{Struktur Data}, C), \\ & (13598025, \text{Hamdan}, \text{Ars. Komputer}, B) \\ & \} \end{aligned}$$

Relasi *MHS* di atas juga dapat ditulis dalam bentuk Tabel:

NIM	Nama	MatKul	Nilai
13598011	Amir	Matematika Diskrit	A
13598011	Amir	Arsitektur Komputer	B
13598014	Santi	Algoritma	D
13598015	Irwan	Algoritma	C
13598015	Irwan	Struktur Data	C
13598015	Irwan	Arsitektur Komputer	B
13598019	Ahmad	Algoritma	E
13598021	Cecep	Algoritma	B
13598021	Cecep	Arsitektur Komputer	B
13598025	Hamdan	Matematika Diskrit	B
13598025	Hamdan	Algoritma	A
13598025	Hamdan	Struktur Data	C
13598025	Hamdan	Arsitektur Komputer	B

- Basisdata (*database*) adalah kumpulan tabel.
- Salah satu model basisdata adalah **model basisdata relasional** (*relational database*).
- Model basisdata ini didasarkan pada konsep relasi *n-ary*.
- Pada basisdata relasional, satu tabel menyatakan satu relasi. Setiap kolom pada tabel disebut **atribut**.
- Daerah asal dari atribut adalah himpunan tempat semua anggota atribut tersebut berada.

Contoh basis data relasional:



- Setiap tabel pada basis data diimplementasikan secara fisik sebagai sebuah *file*.
- Satu baris data pada tabel menyatakan sebuah *record*, dan setiap atribut menyatakan sebuah *field*.
- Secara fisik basisdata adalah kumpulan *file*, sedangkan *file* adalah kumpulan *record*, setiap *record* terdiri atas sejumlah *field*.
- Atribut khusus pada tabel yang mengidentifikasikan secara unik elemen relasi disebut **kunci** (*key*).

- Operasi yang dilakukan terhadap basisdata dilakukan dengan perintah pertanyaan yang disebut *query*.
- Contoh *query*:
 - “tampilkan semua mahasiswa yang mengambil mata kuliah Matematika Diskrit”
 - “tampilkan daftar nilai mahasiswa dengan NIM = 13598015”
 - “tampilkan daftar mahasiswa yang terdiri atas NIM dan mata kuliah yang diambil”
- *Query* terhadap basisdata relasional dapat dinyatakan secara abstrak dengan operasi pada relasi *n-ary*.
- Ada beberapa operasi yang dapat digunakan, diantaranya adalah seleksi, proyeksi, dan join.

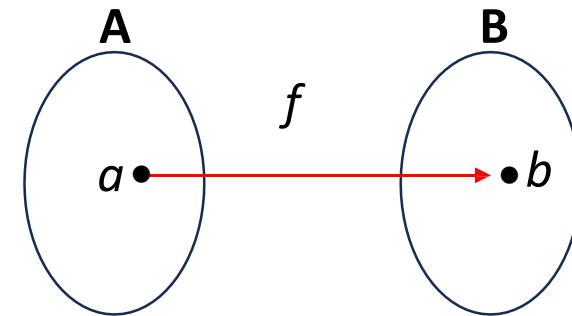
Fungsi

- Misalkan A dan B himpunan. Relasi biner f dari A ke B merupakan suatu fungsi jika *setiap* elemen di dalam A dihubungkan dengan tepat satu elemen di dalam B .

- Jika f adalah fungsi dari A ke B kita menuliskan

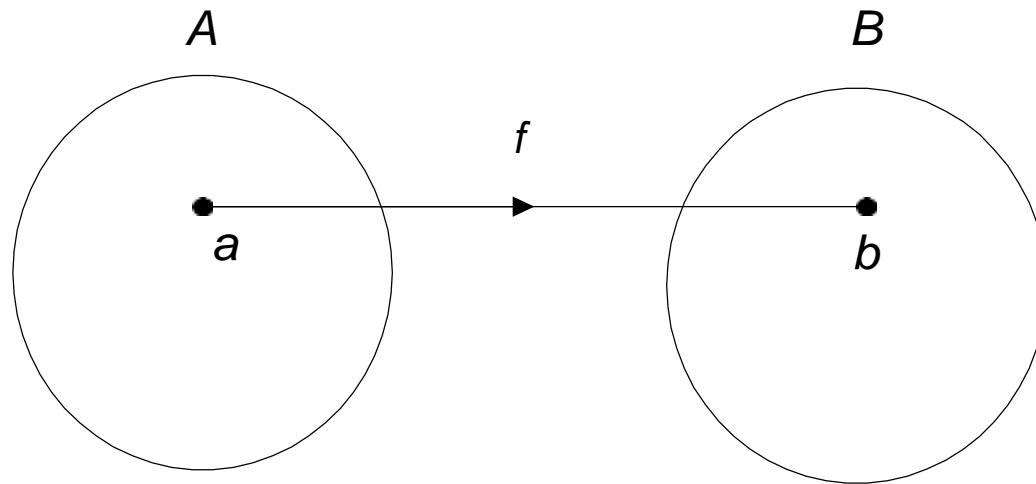
$$f: A \rightarrow B$$

yang artinya f **memetakan** A ke B .



- A disebut **daerah asal** (*domain*) dari f dan B disebut **daerah tujuan** (*codomain*) dari f .
- Nama lain untuk fungsi adalah **pemetaan** atau **transformasi**.
- Kita menuliskan $f(a) = b$ jika elemen a di dalam A dihubungkan dengan elemen b di dalam B .

- Jika $f(a) = b$, maka b dinamakan **bayangan** (*image*) dari a dan a dinamakan **pra-bayangan** (*pre-image*) dari b .
- Himpunan yang berisi semua nilai pemetaan f disebut **jelajah** (*range*) dari f . Perhatikan bahwa jelajah dari f adalah himpunan bagian (mungkin *proper subset*) dari B .



- Fungsi adalah relasi yang khusus:
 1. Tiap elemen di dalam himpunan A harus digunakan oleh prosedur atau kaidah yang mendefinisikan f .
 2. Frasa “dihubungkan dengan tepat satu elemen di dalam B ” berarti bahwa jika $(a, b) \in f$ dan $(a, c) \in f$, maka $b = c$.

- Fungsi dapat dispesifikasikan dalam berbagai bentuk, diantaranya:

1. Himpunan pasangan terurut.
Seperti pada relasi.

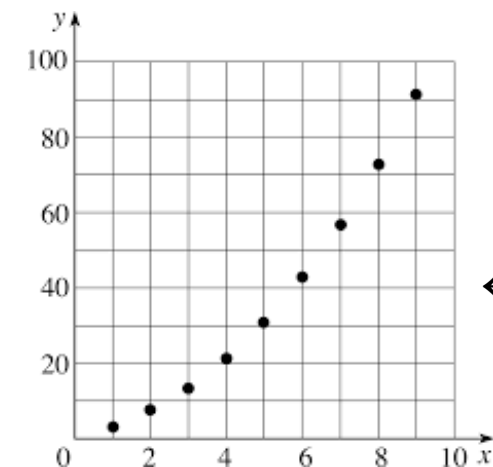
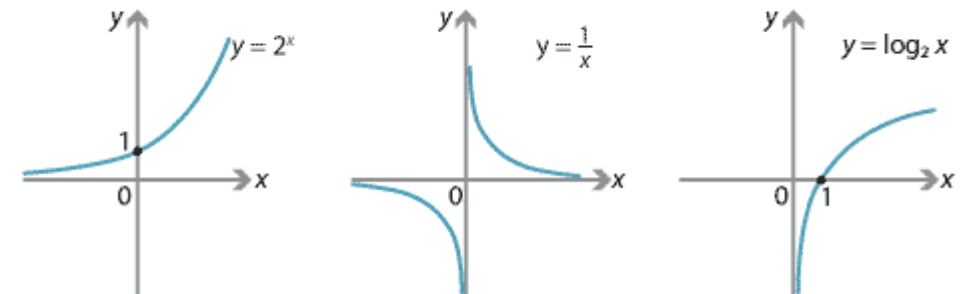
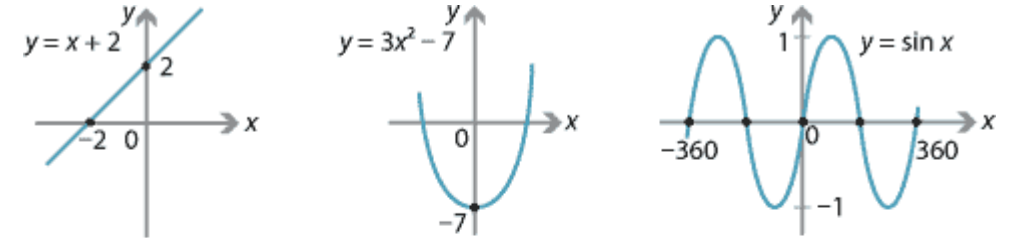
2. Formula pengisian nilai (*assignment*).
Contoh: $f(x) = 2x + 10$, $f(x) = x^2$, dan $f(x) = 1/x$.

3. Kata-kata
Contoh: “ f adalah fungsi yang memetakan jumlah bit 1 di dalam suatu *string* biner”.

4. Kode program (*source code*)
Contoh: Fungsi menghitung $|x|$

```
function abs (x:integer) :integer;
begin
    if x < 0 then
        abs := -x
    else
        abs := x;
end;
```

5. Kurva/grafik dalam bidang kartesian



← fungsi diskrit

Contoh. Relasi

$$f = \{(1, u), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi dari A ke B . Di sini $f(1) = u$, $f(2) = v$, dan $f(3) = w$. Daerah asal dari f adalah A dan daerah tujuan adalah B . Jelajah dari f adalah $\{u, v, w\}$, yang dalam hal ini sama dengan himpunan B .

Contoh. Relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi dari A ke B , meskipun u merupakan bayangan dari dua elemen A . Daerah asal fungsi adalah A , daerah tujuannya adalah B , dan jelajah fungsi adalah $\{u, v\}$.

Contoh. Relasi

$$f = \{(1, u), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3, 4\}$ ke $B = \{u, v, w\}$ bukan fungsi, karena tidak semua elemen A dipetakan ke B .

Contoh. Relasi

$$f = \{(1, u), (1, v), (2, v), (3, w)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi, karena 1 dipetakan ke dua buah elemen B , yaitu u dan v .

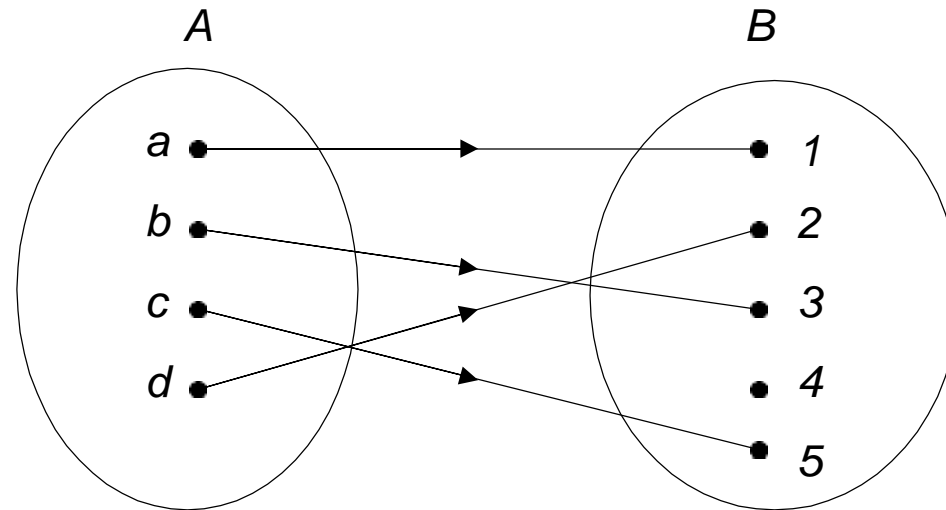
Contoh. Misalkan A adalah himpunan mahasiswa di ITB. Manakah dari pemetaan berikut yang mendefinisikan sebuah fungsi pada himpunan A ?

- (i) Setiap mahasiswa memetakan NIM (Nomor Induk Mahasiswa).
- (ii) Setiap mahasiswa memetakan nomor *handphone*-nya.
- (iii) Setiap mahasiswa memetakan dosen walinya.
- (iv) Setiap mahasiswa memetakan anaknya.

Jawaban:

- (i) Ya, karena setiap mahasiswa hanya mempunyai satu buah NIM.
- (ii) Tidak, karena ada mahasiswa yang mempunyai lebih dari satu nomor HP atau tidak mempunyai HP sama sekali.
- (iii) Ya, karena setiap mahasiswa hanya mempunyai 1 orang dosen wali.
- (iv) Tidak, jika ada mahasiswa yang belum menikah.

- Fungsi f dikatakan **satu-ke-satu** (*one-to-one*) atau **injektif** (*injective*) jika tidak ada dua elemen himpunan A yang memiliki bayangan sama.



Contoh. Relasi

$$f = \{(1, w), (2, u), (3, v)\}$$

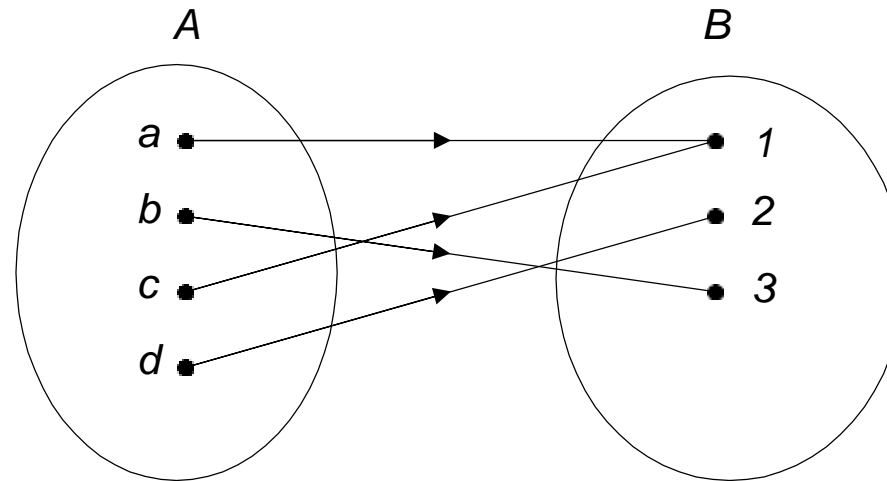
dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w, x\}$ adalah fungsi satu-ke-satu,

Tetapi relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi satu-ke-satu,
karena $f(1) = f(2) = u$.

- Fungsi f dikatakan dipetakan **pada** (*onto*) atau **surjektif** (*surjective*) jika setiap elemen himpunan B merupakan bayangan dari satu atau lebih elemen himpunan A .
- Dengan kata lain seluruh elemen B merupakan jelajah dari f . Fungsi f disebut fungsi pada himpunan B .



Contoh. Relasi

$$f = \{(1, u), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ bukan fungsi pada karena w tidak termasuk jelajah dari f .

Relasi

$$f = \{(1, w), (2, u), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ merupakan fungsi pada karena semua anggota B merupakan jelajah dari f .

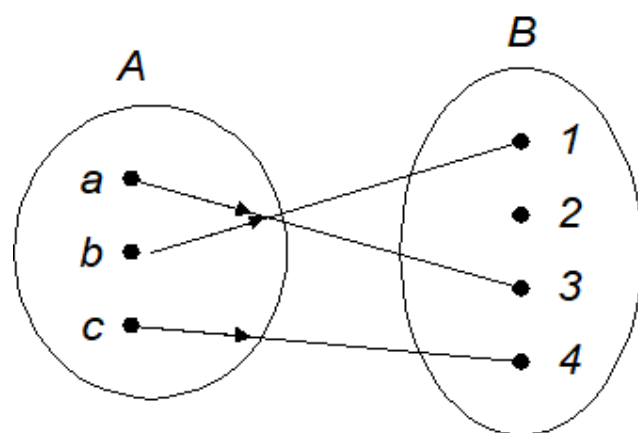
- Fungsi f dikatakan **berkoresponden satu-ke-satu** atau **bijektif** (*bijective*) jika ia fungsi satu-ke-satu dan juga fungsi pada.

Contoh. Relasi

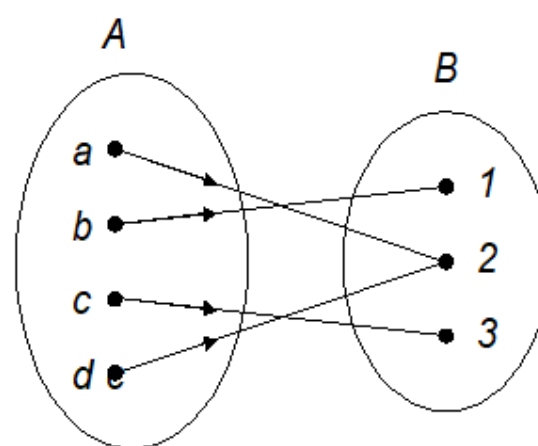
$$f = \{(1, u), (2, w), (3, v)\}$$

dari $A = \{1, 2, 3\}$ ke $B = \{u, v, w\}$ adalah fungsi yang berkoresponden satu-ke-satu, karena f adalah fungsi satu-ke-satu maupun fungsi pada.

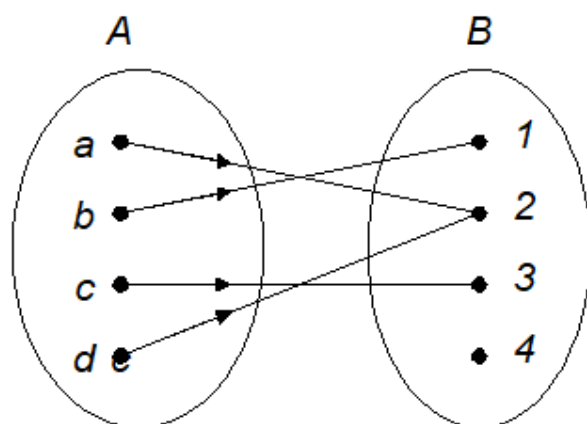
Fungsi satu-ke-satu,
bukan pada



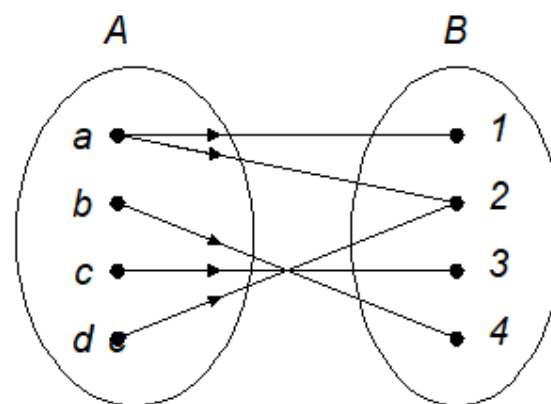
Fungsi pada,
bukan satu-ke-satu



Bukan fungsi satu-ke-satu
maupun pada



Bukan fungsi



Beberapa Fungsi Khusus

1. Fungsi *Floor* dan *Ceiling*

Misalkan x adalah bilangan riil, berarti x berada di antara dua bilangan bulat.

Fungsi *floor* dari x :

$\lfloor x \rfloor$ menyatakan nilai bilangan bulat terbesar yang lebih kecil atau sama dengan x

Fungsi *ceiling* dari x :

$\lceil x \rceil$ menyatakan bilangan bulat terkecil yang lebih besar atau sama dengan x

Dengan kata lain, fungsi *floor* membulatkan x ke bawah, sedangkan fungsi *ceiling* membulatkan x ke atas.

Contoh. Beberapa contoh nilai fungsi *floor* dan *ceiling*:

$$\lfloor 3.5 \rfloor = 3$$

$$\lceil 3.5 \rceil = 4$$

$$\lfloor 0.5 \rfloor = 0$$

$$\lceil 0.5 \rceil = 1$$

$$\lfloor 4.8 \rfloor = 4$$

$$\lceil 4.8 \rceil = 5$$

$$\lfloor -0.5 \rfloor = -1$$

$$\lceil -0.5 \rceil = 0$$

$$\lfloor -3.5 \rfloor = -4$$

$$\lceil -3.5 \rceil = -3$$

Contoh. Di dalam komputer, data dikodekan dalam untaian *byte*, satu *byte* terdiri atas 8 bit. Jika panjang data 125 bit, maka jumlah *byte* yang diperlukan untuk merepresentasikan data adalah $\lceil 125/8 \rceil = 16$ *byte*. Perhatikanlah bahwa $16 \times 8 = 128$ bit, sehingga untuk *byte* yang terakhir perlu ditambahkan 3 bit ekstra agar satu *byte* tetap 8 bit (bit ekstra yang ditambahkan untuk menggenapi 8 bit disebut *padding bits*).

2. Fungsi modulo

Misalkan a adalah sembarang bilangan bulat dan m adalah bilangan bulat positif.

$a \bmod m$ memberikan sisa pembagian bilangan bulat bila a dibagi dengan m

$a \bmod m = r$ sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$.

Contoh . Beberapa contoh fungsi modulo

$$25 \bmod 7 = 4$$

$$15 \bmod 4 = 3$$

$$3612 \bmod 45 = 12$$

$$0 \bmod 5 = 0$$

3. Fungsi Faktorial

$$n! = \begin{cases} 1 & , n = 0 \\ 1 \times 2 \times \cdots \times (n-1) \times n & , n > 0 \end{cases}$$

4. Fungsi Eksponensial

$$a^n = \begin{cases} 1 & , n = 0 \\ \underbrace{a \times a \times \cdots \times a}_n & , n > 0 \end{cases}$$

Untuk kasus perpangkatan negatif,

$$a^{-n} = \frac{1}{a^n}$$

5. Fungsi Logaritmik

Fungsi logaritmik berbentuk

$$y = {}^a \log x \quad \Leftrightarrow \quad x = a^y$$

Fungsi Rekursif

- Fungsi f dikatakan fungsi rekursif jika definisi fungsinya mengacu pada dirinya sendiri.

Contoh: $n! = 1 \times 2 \times \dots \times (n - 1) \times n = (n - 1)! \times n.$

$$n! = \begin{cases} 1 & , n = 0 \\ n \times (n - 1)! & , n > 0 \end{cases}$$

Fungsi rekursif disusun oleh dua bagian:

(a) *Basis*

Bagian yang berisi nilai awal yang tidak mengacu pada dirinya sendiri. Bagian ini juga sekaligus menghentikan definisi rekursif.

(b) *Rekurens*

Bagian ini mendefinisikan argumen fungsi dalam terminologi dirinya sendiri. Setiap kali fungsi mengacu pada dirinya sendiri, argumen dari fungsi harus lebih dekat ke nilai awal (basis).

- Contoh definisi rekursif dari faktorial:

(a) basis:

$$n! = 1 \quad , \text{ jika } n = 0$$

(b) rekurens:

$$n! = n \times (n - 1)! \quad , \text{ jika } n > 0$$

5! dihitung dengan langkah berikut:

$$(1) \ 5! = 5 \times 4! \quad (\text{rekurens})$$

$$(2) \quad \quad \quad 4! = 4 \times 3!$$

$$(3) \quad \quad \quad \quad 3! = 3 \times 2!$$

$$(4) \quad \quad \quad \quad \quad 2! = 2 \times 1!$$

$$(5) \quad \quad \quad \quad \quad \quad 1! = 1 \times 0!$$

$$(6) \quad \quad \quad \quad \quad \quad \quad 0! = 1$$

$$(6') \ 0! = 1$$

$$(5') \ 1! = 1 \times 0! = 1 \times 1 = 1$$

$$(4') \ 2! = 2 \times 1! = 2 \times 1 = 2$$

$$(3') \ 3! = 3 \times 2! = 3 \times 2 = 6$$

$$(2') \ 4! = 4 \times 3! = 4 \times 6 = 24$$

$$(1') \ 5! = 5 \times 4! = 5 \times 24 = 120$$

Jadi, $5! = 120$.

Relasi Kesetaraan

DEFINISI. Relasi R pada himpunan A disebut **relasi kesetaraan** (*equivalence relation*) jika ia refleksif, setangkup dan menghantar.

- Secara intuitif, di dalam relasi kesetaraan, dua benda berhubungan jika keduanya memiliki beberapa sifat yang sama atau memenuhi beberapa persyaratan yang sama.
- Dua elemen yang dihubungkan dengan relasi kesetaraan dinamakan **setara** (*equivalent*).

- **Contoh:** Misalkan A = himpunan mahasiswa dan R adalah relasi pada A sedemikian sehingga $(a, b) \in R$ jika a satu angkatan dengan b .

→ R refleksif: setiap mahasiswa seangkatan dengan dirinya sendiri

→ R setangkup: jika a seangkatan dengan b , maka b pasti seangkatan dengan a .

→ R menghantar: jika a seangkatan dengan b dan b seangkatan dengan c , maka pastilah a seangkatan dengan c .

Dengan demikian, R adalah relasi kesetaraan.

Relasi Pengurutan Parsial

DEFINISI. Relasi R pada himpunan S dikatakan **relasi pengurutan parsial** (*partial ordering relation*) jika ia refleksif, tolak-setangkup, dan menghantar.

Himpunan S bersama-sama dengan relasi R disebut **himpunan terurut secara parsial** (*partially ordered set*, atau *poset*), dan dilambangkan dengan (S, R) .

Contoh. Relasi \geq pada himpunan bilangan bulat positif adalah relasi pengurutan parsial.

Alasan:

Relasi \geq refleksif, karena $a \geq a$ untuk setiap bilangan bulat a ;

Relasi \geq tolak-setangkup, karena jika $a \geq b$ dan $b \geq a$, maka $a = b$;

Relasi \geq menghantar, karena jika $a \geq b$ dan $b \geq c$ maka $a \geq c$.

Contoh : Relasi “habis membagi” pada himpunan bilangan bulat adalah relasi pengurutan parsial.

Alasan: relasi “habis membagi” bersifat refleksif, tolak-setangkup, dan menghantar.

- Secara intuitif, di dalam relasi pengurutan parsial, dua buah benda saling berhubungan jika salah satunya
 - lebih kecil (lebih besar) daripada,
 - atau lebih rendah (lebih tinggi) daripada lainnya

menurut sifat atau kriteria tertentu.

- Istilah pengurutan menyatakan bahwa benda-benda di dalam himpunan tersebut diurutkan berdasarkan sifat atau kriteria tersebut.
- Ada juga kemungkinan dua buah benda di dalam himpunan tidak berhubungan dalam suatu relasi pengurutan parsial. Dalam hal demikian, kita tidak dapat membandingkan keduanya sehingga tidak dapat diidentifikasi mana yang lebih besar atau lebih kecil.
- Itulah alasan digunakan istilah pengurutan parsial atau pengurutan tak-lengkap

Klosur Relasi (*closure of relation*)

- **Contoh kasus:** Relasi $R = \{(1, 1), (1, 3), (2, 3), (3, 2)\}$ pada himpunan $A = \{1, 2, 3\}$ tidak bersifat refleksif.
- Bagaimana membuat relasi refleksif yang sesedikit mungkin dan mengandung R ?

- Tambahkan $(2, 2)$ dan $(3, 3)$ ke dalam R (karena dua elemen relasi ini yang belum terdapat di dalam R)
- Relasi baru, S , mengandung R , yaitu

$$S = \{(1, 1), (1, 3), \mathbf{(2, 2)}, (2, 3), (3, 2), \mathbf{(3, 3)}\}$$

- Relasi S disebut **klosur refleksif** (*reflexive closure*) dari R .

- **Contoh kasus 2:** Relasi $R = \{(1, 3), (1, 2), (2, 1), (3, 2), (3, 3)\}$ pada himpunan $A = \{1, 2, 3\}$ tidak setangkup.
- Bagaimana membuat relasi setangkup yang sesedikit mungkin dan mengandung R ?

- Tambahkan $(3, 1)$ dan $(2, 3)$ ke dalam R
(karena dua elemen relasi ini yang belum terdapat di dalam S agar S menjadi setangkup).
- Relasi baru, S , mengandung R :

$$S = \{(1, 3), (3, 1), (1, 2), (2, 1), (3, 2), (2, 3), (3, 3)\}$$

- Relasi S disebut **klosur setangkup** (*symmetric closure*) dari R .

- Misalkan R adalah relasi pada himpunan A . R dapat memiliki atau tidak memiliki sifat **P**, seperti refleksif, setangkup, atau menghantar.
- Jika terdapat relasi S dengan sifat **P** yang mengandung R sedemikian sehingga S adalah himpunan bagian dari setiap relasi dengan sifat P yang mengandung R ,
- maka S disebut **klosur** (*closure*) atau tutupan dari R .

Klosur Refleksif

- Misalkan R adalah sebuah relasi pada himpunan A .
- Klosur refleksif dari R adalah $R \cup \Delta$, yang dalam hal ini
$$\Delta = \{(a, a) \mid a \in A\}.$$

- **Contoh:** Misalkan $R = \{(1, 1), (1, 3), (2, 3), (3, 2)\}$ adalah relasi pada $A = \{1, 2, 3\}$

maka

$$\Delta = \{(1, 1), (2, 2), (3, 3)\},$$

sehingga klosur refleksif dari R adalah

$$\begin{aligned} R \cup \Delta &= \{(1, 1), (1, 3), (2, 3), (3, 2)\} \cup \{(1, 1), (2, 2), (3, 3)\} \\ &= \{(1, 1), (1, 3), (2, 2), (2, 3), (3, 2), (3, 3)\} \end{aligned}$$

Klosur setangkup

- Misalkan R adalah sebuah relasi pada himpunan A .
- Klosur setangkup dari R adalah $R \cup R^{-1}$, dengan $R^{-1} = \{(b, a) \mid (a, b) \in R\}$.

- **Contoh:** Misalkan $R = \{(1, 3), (1, 2), (2, 1), (3, 2), (3, 3)\}$ adalah relasi pada $A = \{1, 2, 3\}$,

maka

$$R^{-1} = \{(3, 1), (2, 1), (1, 2), (2, 3), (3, 3)\}$$

sehingga klosur setangkup dari R adalah

$$\begin{aligned} R \cup R^{-1} &= \{(1, 3), (1, 2), (2, 1), (3, 2), (3, 3)\} \cup \\ &\quad \{(3, 1), (2, 1), (1, 2), (2, 3), (3, 3)\} \\ &= \{(1, 3), (3, 1), (1, 2), (2, 1), (3, 2), (2, 3), (3, 3)\} \end{aligned}$$

Klosur menghantar

- Pembentukan klosur menghantar lebih sulit daripada dua buah klosur sebelumnya.
- Contoh: $R = \{(1, 2), (1, 4), (2, 1), (3, 2)\}$ adalah relasi $A = \{1, 2, 3, 4\}$.
- R tidak transitif karena tidak mengandung semua pasangan (a, c) sedemikian sehingga (a, b) dan (b, c) di dalam R .
- Pasangan (a, c) yang tidak terdapat di dalam R adalah $(1, 1)$, $(2, 2)$, $(2, 4)$, dan $(3, 1)$.

- Penambahan semua pasangan ini ke dalam R sehingga menjadi

$$S = \{(1, 2), (1, 4), (2, 1), (3, 2), (1, 1), (2, 2), (2, 4), (3, 1)\}$$

tidak menghasilkan relasi yang bersifat menghantar karena, misalnya terdapat $(3, 1) \in S$ dan $(1, 4) \in S$, tetapi $(3, 4) \notin S$.

- Klosur menghantar dari R adalah

$$R^* = R \cup R^2 \cup R^3 \cup \dots \cup R^n$$

- Jika M_R adalah matriks yang merepresentasikan R pada sebuah himpunan dengan n elemen, maka matriks klosur menghantar R^* adalah

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee \dots \vee M_R^{[n]}$$

Contoh. Misalkan $R = \{(1, 1), (1, 3), (2, 2), (3, 1), (3, 2)\}$ adalah relasi pada himpunan $A = \{1, 2, 3\}$. Tentukan klosur menghantar dari R .

Penyelesaian:

Matriks yang merepresentasikan relasi R adalah

$$M_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Maka, matriks klosur menghantar dari R adalah

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]}$$

Karena

$$M_R^{[2]} = M_R \cdot M_R = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \text{ dan } M_R^{[3]} = M_R \cdot M_R^{[2]} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

maka

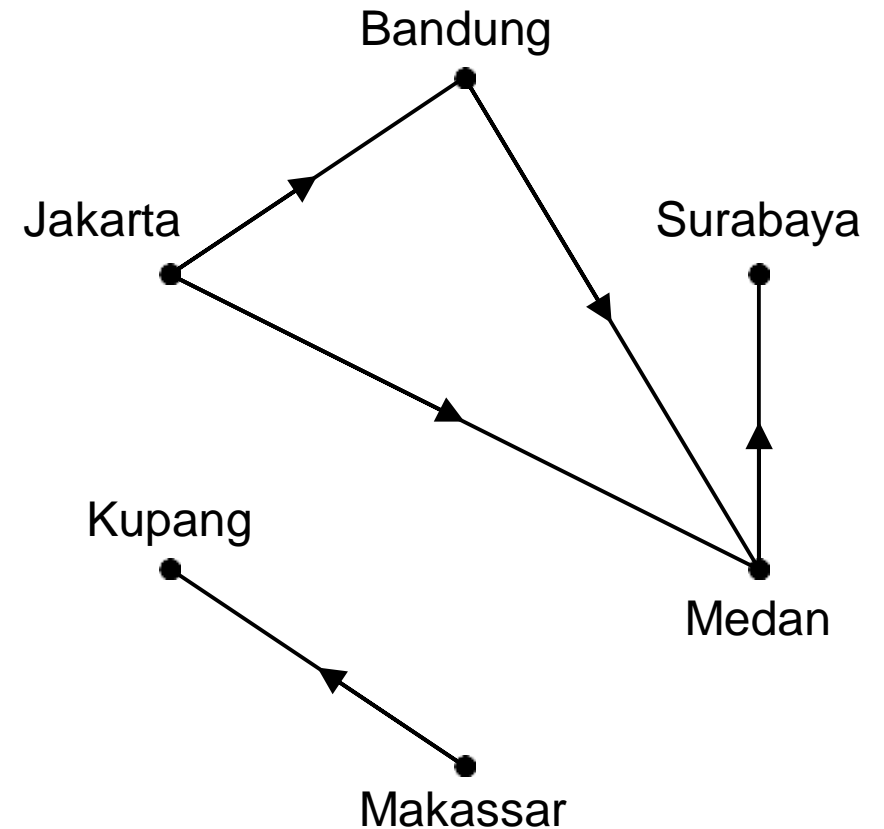
$$M_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Dengan demikian, $R^* = \{(1, 1), (1, 2), (1, 3), (2, 2), (3, 1), (3, 2), (3, 3)\}$

Aplikasi klosur menghantar

- Klosur menghantar menggambarkan bagaimana pesan dapat dikirim dari satu kota ke kota lain baik melalui hubungan komunikasi langsung atau melalui kota antara sebanyak mungkin.

- Misalkan jaringan komputer mempunyai pusat data di Jakarta, Bandung, Surabaya, Medan, Makassar, dan Kupang.
- Misalkan R adalah relasi yang mengandung (a, b) jika terdapat saluran telepon dari kota a ke kota b .



- Karena tidak semua *link* langsung dari satu kota ke kota lain, maka pengiriman data dari Jakarta ke Surabaya tidak dapat dilakukan secara langsung.
- Relasi R tidak menghantar karena ia tidak mengandung semua pasangan pusat data yang dapat dihubungkan (baik *link* langsung atau tidak langsung).
- Klosur menghantar adalah relasi yang paling minimal yang berisi semua pasangan pusat data yang mempunyai link langsung atau tidak langsung dan mengandung R .

