# Relasi n-ary dan Basisdata relasional

# Relasi n-ary

Untuk mempelajari suatu aplikasi relasi yang menarik yang disebut sebagai **database (pangkalan data)**, kita pertama-tama harus meng-generalisasikan konsep relasi biner ke **relasi n-ary**.

**Definisi:** $A_1, A_2, \ldots, A_n$ adalah himpunan. Suatu **relasi n-ary pada himpunan ini** adalah himpunan bagian dari $A_1 \times A_2 \times \ldots \times A_n$.

Himpunan $A_1, A_2, \ldots, A_n$ disebut **domain** dari relasi dan n adalah **derajat (tingkatan)**.

# Relasi n-ary

**Contoh:**

Mis. $R = \{(a, b, c) \mid a = 2b \wedge b = 2c \text{ dengan } a, b, c \in \mathbf{Z}\}$

Berapakah derajat dari R ?

Derajat dari R adalah 3, jadi anggotanya adalah triple.

Apakah domain dari R ?

Domainnya adalah semua bilangan bulat.

Apakah (2, 4, 8) anggota R? Tidak.

Apakah (4, 2, 1) anggota R ? Ya.

# Relasi dan Database

Kita tinjau suatu jenis repesentasi database yang didasarkan pada relasi, yang disebut sebagai **model basisdata relasional.**

Suatu database terdiri dari n-tupel yang disebut **record**, yang terdiri dari **field-field**.

Field ini merupakan **entri** dari n-tupel.

Model data relasional me-representasikan database sebagai relasi *n-ary*, yaitu himpunan dari record.

# Relasi dan Database

**Contoh:** Tinjau database mahasiswa, yang rekord-nya di-representasikan sebagai 4-tupel dengan field **Nama Mahasiswa**, **Nomor Induk**, **Program Studi**, dan **IPK**:

R = {(Ackermann, 231455, CS, 3.88),
(Adams, 888323, Physics, 3.45),
(Chou, 102147, CS, 3.79),
(Goodfriend, 453876, Math, 3.45),
(Rao, 678543, Math, 3.90),
(Stevens, 786576, Psych, 2.99)}

Relasi yang merepresentasikan database disebut juga sebagai “**Tabel**”, karena seringkali ditampilkan dalam bentuk tabel.

# Relasi dan Database

Suatu domain dari relasi *n-ary* disebut **primary key** (kunci primer) jika n-tupel dapat ditentukan secara unik berdasarkan nilainya pada domain ini.

Ini berarti tidak ada dua rekord yang mempunyai nilai sama dari kunci primer yang sama.

Pada contoh diatas, yang manakah diantara field **Nama Mahasiswa**, **Nomor Induk**, **Program Studi**, dan **IPK** bisa dijadikan sebagai kunci primer ?

**Nama Mahasiswa** dan **Nomor Induk** adalah kunci primer, karena tidak ada dua mahasiswa yang memiliki field ini dengan harga sama.

NB: pada database yang sebenarnya, hanya **Nomor Induk** yang dapat dijadikan kunci primer karena nama dua org mhs bisa sama.

# Relasi dan Database

Dalam database, (nilai) kunci primer harus selalu satu, tidak terduplikasi, meskipun rekord baru ditambahkan.

Oleh karena itu, kita harus memakai kunci primer yang diperluas, yang mengandung semua n-tupel yang bisa dimasukkan kedalam database.

**Kombinasi dari domain** dapat juga dipakai untuk mengidentifikasi n-tupel secara unik pada relasi n-ary.

Jika nilai **sekumpulan domain** dipakai untuk menentukan n-tupel dalam relasi, **Perkalian Kartesian** dari domain ini disebut sebagai **kunci komposite**.

# Relasi dan Database

Berbagai jenis **operasi** pada relasi n-ary dapat dipakai untuk membentuk relasi baru.

**Definisi: Proyeksi** $P_{i_1, i_2, \ldots, i_m}$ memetakan n-tupel $(a_1, a_2, \ldots, a_n)$ ke m-tupel $(a_{i_1}, a_{i_2}, \ldots, a_{i_m})$, dimana $m \leq n$.

Dengan kata lain, suatu proyeksi $P_{i_1, i_2, \ldots, i_m}$ mempertahankan m komponen $a_{i_1}, a_{i_2}, \ldots, a_{i_m}$ dari n-tupel dan menghapus (n – m) komponen lainnya.

**Contoh:** Tentukan hasil proyeksi $P_{2,4}$ pada rekord mahasiswa (Stevens, 786576, Psych, 2.99) ?

**Jawaban:** Hasilnya adalah (786576, 2.99).

# Relasi dan Database

Kadangkala proyeksi tidak hanya menghasilkan kolom yang lebih sedikit, tetapi juga **baris yang lebih sedikit**.

Mengapa?

Ada rekord-rekord yang berbeda hanya pada field yang dihapus, dengan demikian rekord ini berharga sama setelah proyeksi, dan tidak ada gunanya me-list rekord yang sama berkali-kali.

# Relasi dan Database

Operasi **join** (penggabungan) dapat dipakai untuk mengkombinasikan dua tabel menjadi satu jika ada field yang identik pada kedua tabel tsb.

**Definisi:** Misalkan R relasi derajat m dan S relasi derajat n. **Join** $J_p(R, S)$ dari relasi R dengan S, dimana $p \leq m$ dan $p \leq n$, adalah relasi derajat $m + n - p$ yang terdiri dari semua $(m + n - p)$-tupel $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$, dimana m-tupel $(a_1, a_2, \ldots, a_{m-p}, c_1, c_2, \ldots, c_p)$ adalah anggota R dan n-tupel $(c_1, c_2, \ldots, c_p, b_1, b_2, \ldots, b_{n-p})$ adalah anggota S.

# Relasi dan Database

Dengan kata lain, untuk membuat Jp(R, S), kita harus menentukan semua anggota R yang p-komponen terakhirnya bersesuaian dengan p-komponen pertama dari anggota S.

Relasi yang baru terdiri dari anggota persesuaian tsb, yang dikombinasikan menjadi tupel-tupel dengan field yang bersesuaian hanya muncul sekali.

# Relasi dan Database

**Contoh:** Tentukan $J_1(Y, R)$, dimana Y mengandung field **Nama Mahasiswa** dan **Tahun Kelahiran**,

Y = {(1978, Ackermann),
(1972, Adams),
(1917, Chou),
(1984, Goodfriend),
(1982, Rao),
(1970, Stevens)},

dan R mengandung rekord mahasiswa spt pada contoh terdahulu ?

# Relasi dan Database

**Jawab:** Relasi hasilnya adalah:

{(1978, Ackermann, 231455, CS, 3.88),
(1972, Adams, 888323, Physics, 3.45),
(1917, Chou, 102147, CS, 3.79),
(1984, Goodfriend, 453876, Math, 3.45),
(1982, Rao, 678543, Math, 3.90),
(1970, Stevens, 786576, Psych, 2.99)}

Karena Y mempunyai dua field dan R mempunyai empat field, relasi $J_1(Y, R)$ memiliki $2 + 4 - 1 = 5$ field.

# Representasi Relasi

## Dengan Matriks Boolean

# Representasi Relasi

Berbagai cara untuk merepresentasikan relasi telah kita ketahui. Selanjutnya kita akan mempelajari lagi dua macam representasi: **matriks Zero-one (Boolean)** dan **graf berarah (directed graphs/digraph)**.

Jika R adalah relasi dari $A = \{a_1, a_2, \ldots, a_m\}$ ke $B = \{b_1, b_2, \ldots, b_n\}$, maka R dapat direpresentasikan dengan matriks Boolean $M_R = [m_{ij}]$, dimana

$m_{ij} = 1$, jika $(a_i, b_j) \in R$, dan

$m_{ij} = 0$, jika $(a_i, b_j) \notin R$.

Ingat bahwa untuk membuat matriks ini, pertama-tama kita harus mendaftar anggota A dan B dalam **urutan yang tertentu**.

# Representasi Relasi

**Contoh:** Tentukan representasi relasi

R = {(2, 1), (3, 1), (3, 2)} sebagai matriks Boolean ?

**Jawab:** Matriks $M_R$ adalah

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

# Representasi Relasi

Bagaimanakah bentuk matriks representasi **relasi pada sebuah himpunan** (relasi dari A ke A) ?

Matriks tersebut akan berbentuk **bujursangkar**.

Bagaimanakah bentuk matriks representasi relasi **refleksif** ?

Elemen **diagonal** matriks $M_{ref}$ ini sama dengan **1**.

$$M_{ref} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & & & 1 \end{bmatrix}$$

# Representasi Relasi

Bagaimanakah bentuk matriks representasi **relasi simetrik**?

Matriks ini simetrik: $M_R = (M_R)^t$.

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

matriks simetrik,
relasi simetrik.

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

matriks non-simetrik,
relasi non-simetrik,.

# Representasi Relasi

Operasi Boolean **join** dan **meet** dapat digunakan untuk menentukan matriks representasi **gabungan** dan **irisan** dua buah relasi.

Untuk mendapatkan **join** dari dua matriks Boolean, kita menerapkan fungsi Boolean “or” ke semua elemen matriks yang bersesuaian.

Untuk mendapatkan **meet** dari dua matriks Boolean, kita menerapkan fungsi Boolean “and” ke semua elemen yang bersesuaian.

# Representasi Relasi

**Contoh:** Misalkan relasi R dan S direpresentasikan oleh matriks-matriks

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

Tetntukan matriks representasi R∪S dan R∩S?

**Jawab:**

$$M_{R \cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad M_{R \cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

# Merepresentasikan Relasi dng Matriks

Bagaimanakah dengan **perkalian Boolean** dari dua matriks Boolean?

Misalkan $A = [a_{ij}]$ matriks Boolean $m \times k$ dan $B = [b_{ij}]$ matriks Boolean $k \times n$.

Maka **perkalian Boolean** dari A dan B, dituliskan sebagai $A \circ B$, adalah matriks $m \times n$ dimana entry ke (i, j) nya $[c_{ij}]$, dimana

$$c_{ij} = (a_{i1} \wedge b_{1j}) \vee (a_{i2} \wedge b_{2i}) \vee \ldots \vee (a_{ik} \wedge b_{kj}).$$

$c_{ij} = 1$ jika dan hanya jika sedikitnya satu dari suku $(a_{in} \wedge b_{nj}) = 1$ untuk suatu nilai n; dan $c_{ij} = 0$ jika sebaliknya.

# Merepresentasikan Relasi dng Matriks

Tinjau matriks Boolean $M_A = [a_{ij}]$, $M_B = [b_{ij}]$ dan $M_C = [c_{ij}]$ yang merepresentasikan relasi A, B, dan C.

**Ingat:** Untuk $M_C = M_A \circ M_B$ kita punya:

$c_{ij} = 1$ jika dan hanya jika sedikitnya satu dari suku $(a_{in} \wedge b_{nj}) = 1$ untuk n tertentu n; $c_{ij} = 0$ jika sebaliknya.

Dari sudut pandang **relasi**, ini berarti bahwa C mengandung sepasang $(x_i, z_j)$ jika dan hanya jika ada elemen $y_n$ sedemikian hingga $(x_i, y_n)$ adalah relasi didalam A dan $(y_n, z_j)$ adalah relasi didalam B.

Sehingga, $C = B \circ A$ (**komposit** dari A dan B).

# Merepresentasikan Relasi dng Matriks

Kita dapatkan aturan berikut:

$$M_{B \circ A} = M_A \circ M_B$$

Dengan kata lain, matriks representasi relasi **komposit** dari relasi A dan B adalah **perkalian Boolean** dari matriks representasi A dengan matriks representasi B.

Dengan analogi kita dapatkan matriks representasi **relasi perpangkatan**:

$M_{R^n} = M_R^{[n]}$ (**Boolean power** ke-n).

# Merepresentasikan Relasi dng Matriks

**Contoh:** Tentukan matriks representasi $R^2$, dimana matriks representasi R adalah

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

**Jawab:** Matriks representasi $R^2$ adalah

$$M_{R^2} = M_R^{[2]} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

# Representasi Relasi dengan Graf Berarah

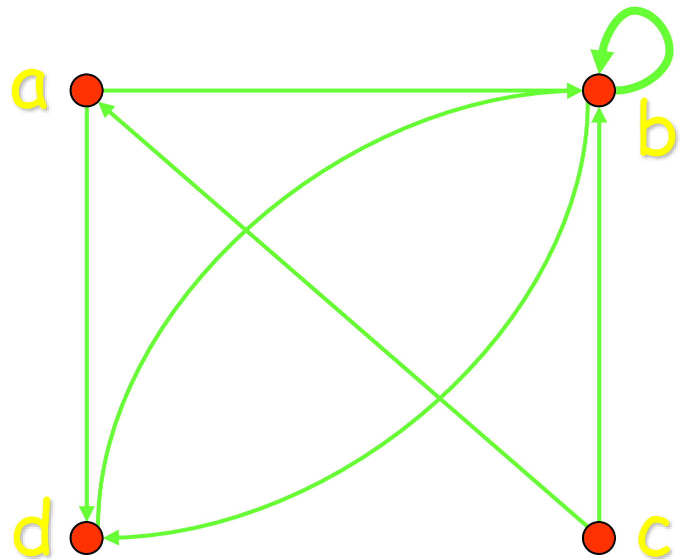
# Merepresentasikan Relasi dng Digraph

**Definisi:** Suatu graf berarah **(directed graph** atau **digraph)**, terdiri dari himpunan verteks/node (simpul) bersama dengan himpunan pasangan berurut E, dari anggota-anggota V, yang disebut sisi (**edge** atau **arcs**). Untuk sisi (a,b), a disebut simpul awal (**initial vertex**) dan b disebut simpul akhir (**terminal vertex**).

Digraf dapat digambarkan dengan anak panah.

# Merepresentasikan Relasi dng Digraph

**Contoh:** Gambarkan digraph dengan V = {a, b, c, d}, E = {(a, b), (a, d), (b, b), (b, d), (c, a), (c, b), (d, b)}.




Sisi berbentuk (b, b) disebut sebagai **loop**.

# Merepresentasikan Relasi dng Digraph

Jelas bahwa relasi R pada himpunan A dapat direpresentasikan dengan digraph dimana A himpunan titik simpulnya dan semua pasangan (a, b)$\in$R sebagai sisi-nya.

Sebaliknya, sebarang digraph dengan simpul V dan sisi E dapat direpresentasikan dengan relasi pada V yang mengandung semua pasangan didalam E.

**Korespondensi satu-ke-satu** antara relasi dan digraph berarti bahwa semua pernyataan mengenai relasi juga berlaku untuk digraph, begitu pula sebaliknya.

# Klosur dari Relasi

# Closure dari Relasi

Apakah **closure** (penutup) dari sebuah relasi ?

**Definisi:** Misalkan R suatu relasi pada himpunan A. Boleh jadi R memiliki atau tidak memiliki **sifat P**, seperti refleksivitas, simetri atau transitivitas. Jika ada relasi S dengan sifat P yang mengandung R sedemikian hingga S himpunan bagian dari semua relasi dengan sifat P yang mengandung R, maka S disebut sebagai closure dari R terhadap P.

**Closure dari suatu relasi terhadap suatu sifat bisa jadi tidak ada.**

# Closure dari Relasi : closure refleksif

**Contoh I:** Tentukan **closure refleksif** dari relasi R = {(1, 1), (1, 2), (2, 1), (3, 2)} pada himpunan
A = {1, 2, 3}.

**Jawab:** Kita tahu bahwa setiap relasi refleksif pada A harus mengandung (1, 1), (2, 2), and (3, 3).

Dengan menambahkan (2,2) dan (3,3) pada R, kita mendapatkan relasi refleksif pada S yang diberikan oleh S = {(1, 1), (1, 2), (2, 1), (2, 2), (3, 2), (3, 3)}.

S refleksif, mengandung R, dan terkandung dalam setiap relasi refleksif yang mengandung R.

Maka, S adalah **closure refleksif** dari R.

# Closure dari Relasi: closure simetrik

**Contoh II:** tentukan **closure simetrik** dari relasi

$R = \{(a, b) \mid a > b\}$ pada himpunan bilangan bulat positif.

**Jawab:** Closur simetrik dari R diberikan oleh

$$R \cup R^{-1} = \{(a, b) \mid a > b\} \cup \{(b, a) \mid a > b\}$$
$$= \{(a, b) \mid a \neq b\}$$

# Closure dari Relasi: closure transitif

**Contoh III:** Tentukan **closure transitif** dari relasi R = {(1, 3), (1, 4), (2, 1), (3, 2)} pada himpunan A = {1, 2, 3, 4}.

**Jawab:** Perdefinisi, R akan transitif, jika untuk semua pasangan (a, b) dan (b, c) didalam R, ada pula pasangan (a, c) didalam R.

Jika kita menambahkan pasangan yang hilang, (1, 2), (2, 3), (2, 4), dan (3, 1), apakah R menjadi transitif?

**Tidak**, karena relasi yg diperluas R mengandung (3, 1) dan (1, 4), tetapi tidak mengandung (3, 4).

Dengan menambahkan anggota baru ke R, kita juga menambahkan **persyaratan baru** untuk sifat transitif nya. Untuk memecahkan permasalahan ini, kita harus melihat **lintasan dalam digraphs**.

# Closure dari Relasi

Bayangkan suatu relasi R yang merepresentasikan semua **jalur kereta api**.

Misalnya, jika (Jakarta, Bandung) anggota R, maka ada jalur kereta langsung dari Jakarta ke Bandung.

Jika R mengandung pasangan berurut (Jakarta, Bandung) dan (Bandung, Tasikmalaya), ada jalur tak langsung dari Jakarta ke Tasikmalaya.

Karena adanya jalur tak langsung, tidak mungkin menentukan kota apa saja yang dihubungkan dengan jalur kereta api, hanya dengan melihat R.

Closure transitif dari R mengandung tepat pasangan-pasangan kota yang terhubung ; **baik yang langsung maupun yang tidak langsung.**

# Closure dari Relasi

**Definisi: Lintasan** dari a ke b dalam digraph G adalah deretan satu atau lebih sisi

$$(x_0, x_1), (x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n)$$

didalam G, dimana $x_0 = a$ dan $x_n = b$.

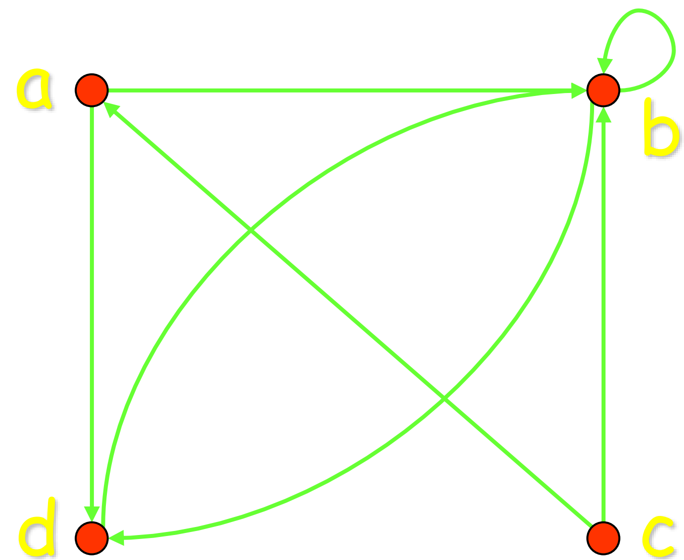
Dengan kata lain, suatu lintasan adalah **deretan sisi** dimana simpul akhir dari suatu sisi sama dengan simpul awal dari sisi berikutnya didalam lintasan.

Lintasan ini dituliskan sebagai $x_0, x_1, x_2, \ldots, x_n$ dan memiliki **panjang** n.

Lintasan yang berawal dan berakhir pada simpul yang sama disebut sebagai **sirkit (circuit)** atau **siklus (cycle)**.

# Closure dari Relasi

**Contoh:** Perhatikan graf berikut:




Apakah **c,a,b,d,b** suatu lintasan? Ya.

apakah **d,b,b,b,d,b,d** suatu siklus? Ya.

Adakah siklus yang mengandung **c**? Tidak.

# Lebih lanjut ttg closure transitif

# Closure dari Relasi

Karena ada korespondensi satu-ke-satu antara graf dng relasi, kita bisa menerapkan definisi dari lintasan didalam graf kedalam relasi:

**Definisi:** Akan ada suatu lintasan dari a ke b didalam relasi R, jika ada deretan elemen a, $x_1$, $x_2$, …, $x_{n-1}$, b dengan $(a, x_1) \in R$, $(x_1, x_2) \in R$, …, dan $(x_{n-1}, b) \in R$.

**Teorema:** Misalkan R relasi pada himpunan A (dimana $|A|=n$). Ada lintasan dari a ke b jika dan hanya jika $(a, b) \in R^n$.

# Closure dari Relasi

Berdasarkan contoh lintasan kereta api terdahulu, *closure* transitif suatu relasi terdiri dari pasangan simpul pada digraf bersangkutan yang dihubungkan dengan suatu lintasan.

**Definisi:** Misalkan R relasi pada A. Relasi konektivitas R* terdiri atas pasangan (a,b) sedemikian hingga ada lintasan antara a dan b didalam R.

Kita tahu bahwa $R^n$ terdiri atas pasangan (a,b) sedemikian hingga a dan b dihubungkan oleh lintasan dengan panjang n.

Oleh karena itu, relasi konektivitas R* adalah gabungan dari $R^n$ untuk semua bilangan positif n:

$$R^* = \bigcup_{n=1}^{\infty} R^n = R^1 \cup R^2 \cup R^3 \cup \ldots$$

# Closure dari Relasi

**Teorema:** Closure transitif dari relasi R sama dengan konektivitas dari relasi R*.

Bagaimana cara menghitung R* ?

**Lemma:** Misalkan *A* suatu himpunan dengan *n* anggota, dan misalkan *R* adalah suatu relasi didalam A. Jika ada lintasan dari *a* ke *b* didalam *R*, maka ada lintasan yang panjangnya tidak melebihi *n*.

Selanjutnya, jika $a \neq b$ dan ada lintasan dari *a* ke *b* didalam R, maka ada lintasan yang panjangnya tidak melebihi (*n*-1).

# Closure dari Relasi

Lemma ini didasarkan pada pengamatan bahwa, jika suatu lintasan dari a ke b melewati sebarang simpul lebih dari sekali, maka akan ada sedikitnya satu buah **siklus.**

Siklus ini bisa dihilangkan dari lintasan, dan lintasan hasilnya masih menghubungkan a dengan b.

**Torema:** Relasi R pada himpunan A dengan *n* anggota, memiliki closure transitif R* yang diberikan oleh:

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

Dinyatakan sebagai matriks relasi:

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

# Closure dari Relasi

Pemecahan **Contoh III** ditentukan dengan mencari **closure transitif** dari relasi R = {(1, 3), (1, 4), (2, 1), (3, 2)} pada himpunan A = {1, 2, 3, 4}.

R dapat direpresentasikan dengan matriks $M_R$:

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

# Closure dari Relasi

$$M_R = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad M_R^{[2]} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_R^{[3]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad M_R^{[4]} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M_{R^*} = M_R \vee M_R^{[2]} \vee M_R^{[3]} \vee M_R^{[4]} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

# Closure dari Relasi

**Jawaban:** Closure transitif dari relasi R = {(1, 3), (1, 4), (2, 1), (3, 2)} pada himpunan A = {1, 2, 3, 4} adalah relasi

{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 1), (3, 2), (3, 3), (3, 4)}

# Algoritma 1.

**ALGORITHM 1 A Procedure for Computing the Transitive Closure.**

**procedure** *transitive closure* ($\mathbf{M}_R$ : zero–one $n \times n$ matrix)
A := $\mathbf{M}_R$
B := A
**for** $i$ := 2 **to** $n$
    A := A $\odot$ $\mathbf{M}_R$
    B := B $\vee$ A
**return** B{B is the zero–one matrix for $R^*$}

- Kompleksitas $O(n^4)$