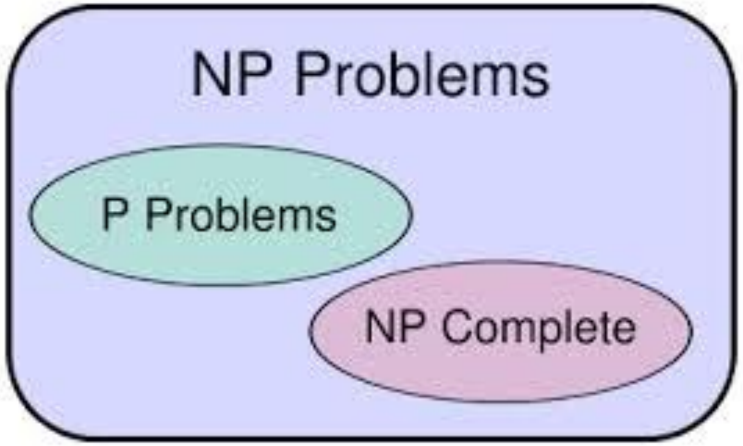


Teori P, NP, dan NP-Complete
(Bagian 2)

Bahan Kuliah IF2211 Strategi Algoritma

Oleh: Rinaldi Munir
Program Studi Teknik Informatika ITB

Ikhtisar

- P Problem
- NP Problem
- NP-Complete Problem
- NP-hard

P Problems

- *P Problems* adalah himpunan semua persoalan keputusan yang dapat dipecahkan oleh algoritma dengan kebutuhan waktu polinom.

- Semua persoalan keputusan yang dapat diselesakan dalam waktu polinom adalah anggota himpunan *P*.

  Contoh:

  1. Persoalan *searching*
  2. Persoalan tes bilangan prima.
  3. Persoalan uji kesamaan dua buah matriks

- Apakah *Travelling Salesperson Decision Problem* (TSDP) termasuk *P Problems*?

- Meskipun belum ada orang yang menemukan algoritma TSDP dalam waktu polinom, namun tidak seorang pun dapat membuktikan bahwa TSDP tidak dapat dipecahkan dalam waktu polinom.

- Ini berarti TSDP *mungkin* dapat dimasukkan ke dalam *P Problems*.

NP Problems

- *NP = non-deterministic polynomial*
- *Non-deterministic polynomial-time algorithm* adalah algoritma non-deterministik yang tahap verifikasinya dapat dilakukan dalam waktu polinomial.

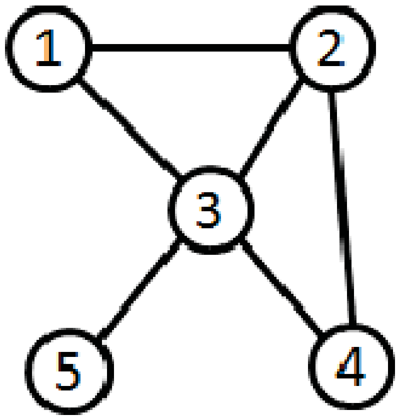
- Verifikasi dalam waktu polinom artinya:

  - Jika diberikan sebuah kandidat jawaban, kita dapat memeriksa apakah jawaban itu benar atau salah dalam waktu polinom.

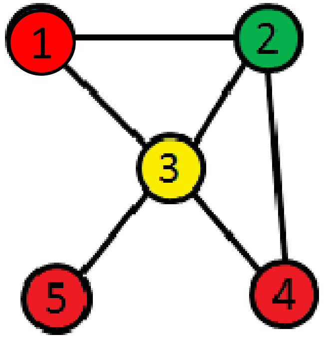
  - Ini berbeda dengan menemukan solusi persoalan dalam waktu polinom

- *NP Problems* adalah himpunan persoalan keputusan yang dapat diselesaikan oleh *non-deterministic polynomial-time algorithm*

- Contoh verifikasi pada pewarnaan graf: Periksa apakah graf G= (V, E) di bawah ini dapat diwarnai dengan tiga warna sehingga tidak ada dua simpul bertetangga berwarna sama



<u>Verifikasi jawaban</u>: *Assignment* simpul-simpul dengan tiga warna adalah benar



Memeriksa apakah dua simpul bertetangga berwarna sama kompleksitasnya O(|E|) → polynomial

- Contoh verifikasi pada *Satisfiable Problem* (SAT):

$X = \{x_1, x_2, x_3\}$ $\quad$ $C = (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2) \wedge (x_1 \vee \bar{x}_2)$

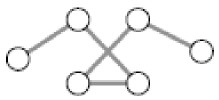
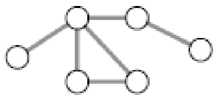
The truth assignment $x_1=1, x_2=1, x_3=1$ satisfies C.

<u>Verifikasi jawaban</u>:
Assignment C dengan nilai-nilai kebenaran tersebut menghasilkan C = true
→ O(1)

- Sayangnya, <u>menemukan</u> solusi untuk kedua persoalan di atas sangat sukar

Some problems in NP

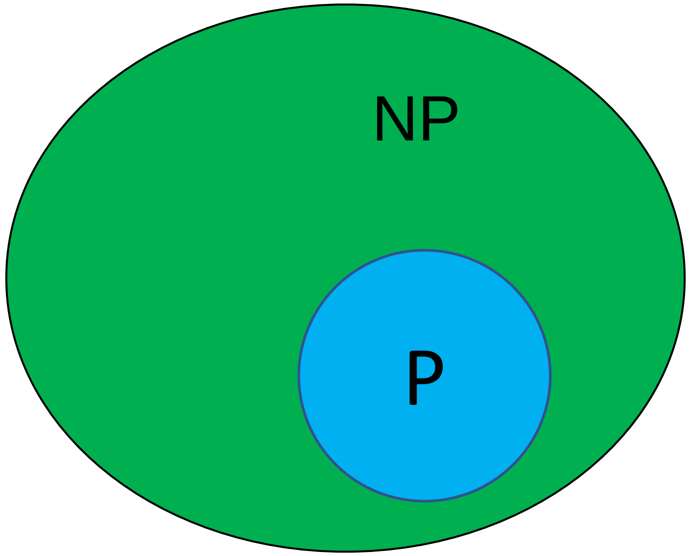
NP. Decision problems for which there exists a poly-time certifier.

problem	description	poly-time algorithm	yes	no
L-SOLVE	Is there a vector x that satisfies $Ax = b$?	Gauss–Edmonds elimination	$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 4 & -2 \\ 0 & 3 & 15 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ 36 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
COMPOSITES	Is x composite ?	Agrawal–Kayal–Saxena	51	53
FACTOR	Does x have a nontrivial factor less than y ?	???	(56159, 50)	(55687, 50)
SAT	Given a CNF formula, does it have a satisfying truth assignment?	???	$\neg x_1 \lor x_2 \lor \neg x_3$ $x_1 \lor \neg x_2 \lor x_3$ $\neg x_1 \lor \neg x_2 \lor x_3$	$\neg x_2$ $x_1 \lor x_2$ $\neg x_1 \lor x_2$
HAMILTON-PATH	Is there a simple path between u and v that visits every node?	???		

- Contoh-contoh persoalan NP:
  1. TSDP, sebab jika diberikan solusi sebagai sebuah terkaan tur, maka dibutuhkan O(n) untuk memverifikasi solusi.
  2. *Integer Knapsack Decision problem*

- P adalah himpunan bagian dari NP, atau $P \subseteq NP$, karena setiap persoalan keputusan yang kompleksitas waktunya polinomial solusinya juga dapat diverifikasi dalam waktu polinom. Di dalam persoalan P tidak ada tahap menerka.

Diagram Venn:



$P \subseteq NP$

- P $\subseteq$ NP mengindikasikan dua hal:

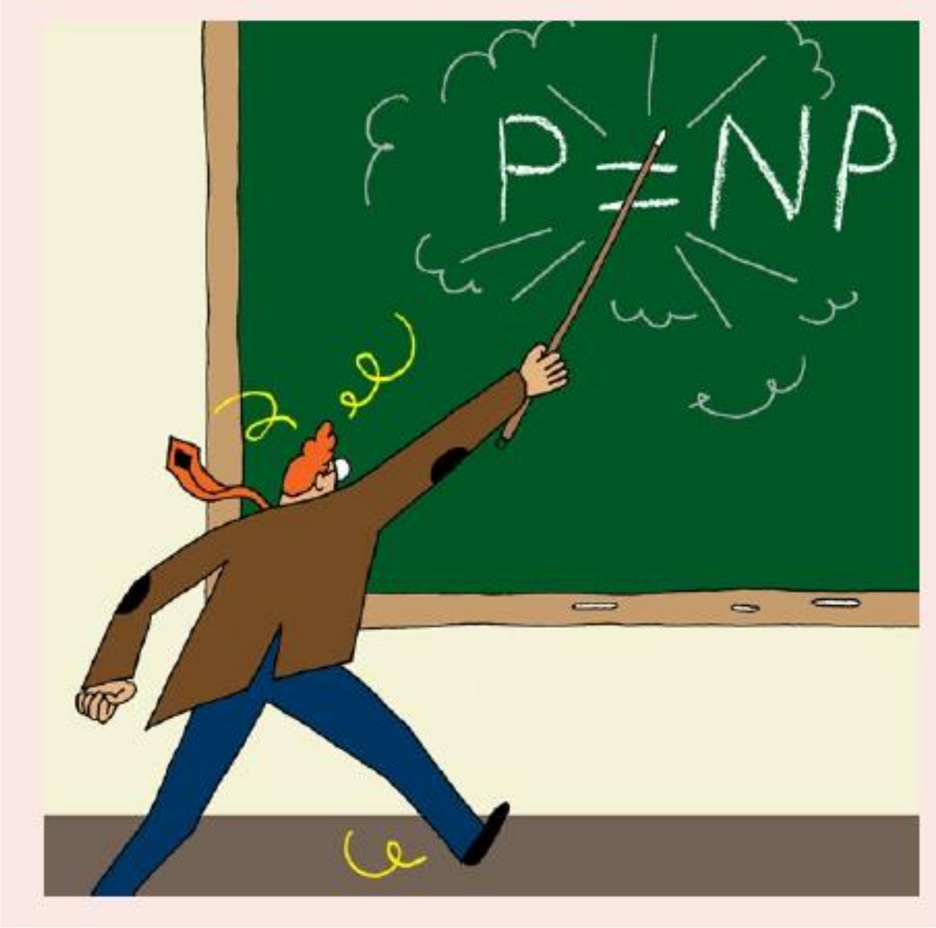
  (i) P = NP   (*improper subset*), atau

  (ii) P ≠ NP   (atau P $\subset$ NP, *proper subset*)

- Tidak seorangpun pernah membuktikan bahwa P $\neq$ NP atau P = NP.

- Pertanyaan apakah P = NP adalah salah satu pertanyaan penting dalam ilmu komputer.

- Pertanyaan ini sangat penting sebab, seperti telah disebutkan sebelumnya, kebanyakan persoalan keputusan adalah NP.



- Karena itu, jika P = NP, maka betapa banyak persoalan keputusan yang dapat dipecahkan secara sangkil (efisien) dengan algoritma yang kebutuhan waktunya polinom.

- Namun kenyataannya, banyak ahli yang telah gagal menemukan algoritma waktu-polinom untuk persoalan NP.

- Karena itu, cukup aman kalau kita mengasumsikan saat ini bahwa $P \neq NP$

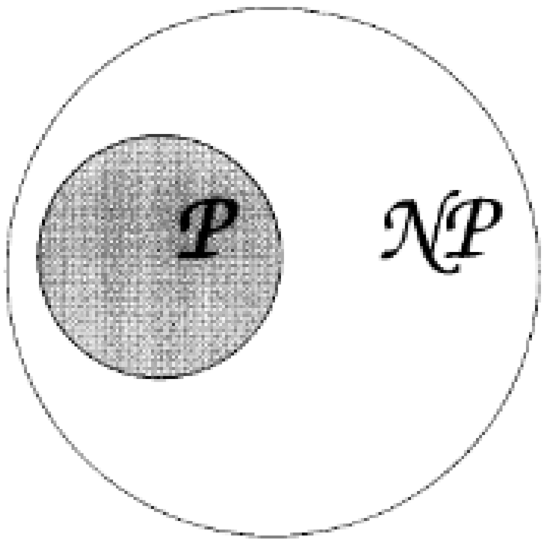
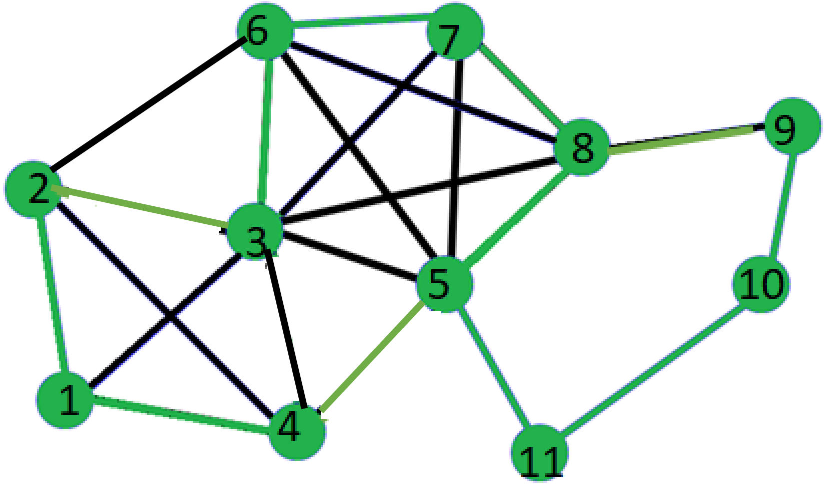


Figure 11.1 Commonly believed relationship between $\mathcal{P}$ and $\mathcal{NP}$

- Adakah persoalan yang tidak termasuk ke dalam NP? Ada, yaitu persoalan *unsolvale*. Contohnya *halting problem*.

- The **P versus NP problem** is a major unsolved problem in computer science. Informally, it asks whether every problem whose solution can be quickly verified by a computer can also be quickly solved by a computer. It was introduced in 1971 by Stephen Cook in his seminal paper "The complexity of theorem proving procedures"[2] and is considered by many to be the most important open problem in the field.[3] It is one of the seven Millennium Prize Problems selected by the Clay Mathematics Institute to carry a US$1,000,000 prize for the first correct solution. (Sumber: Wikipedia)

- Contoh: Persoalan sirkuit Hamilton

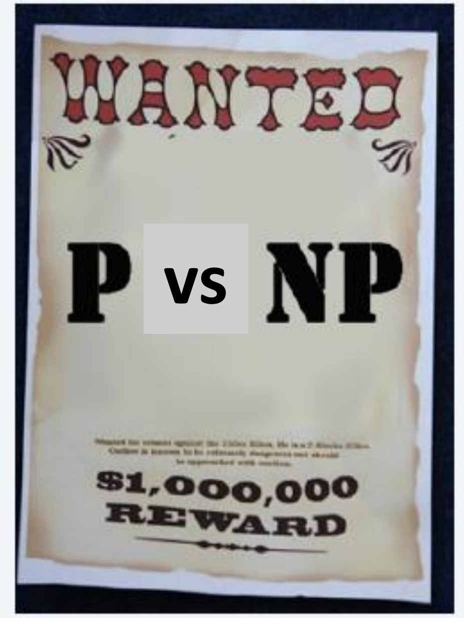


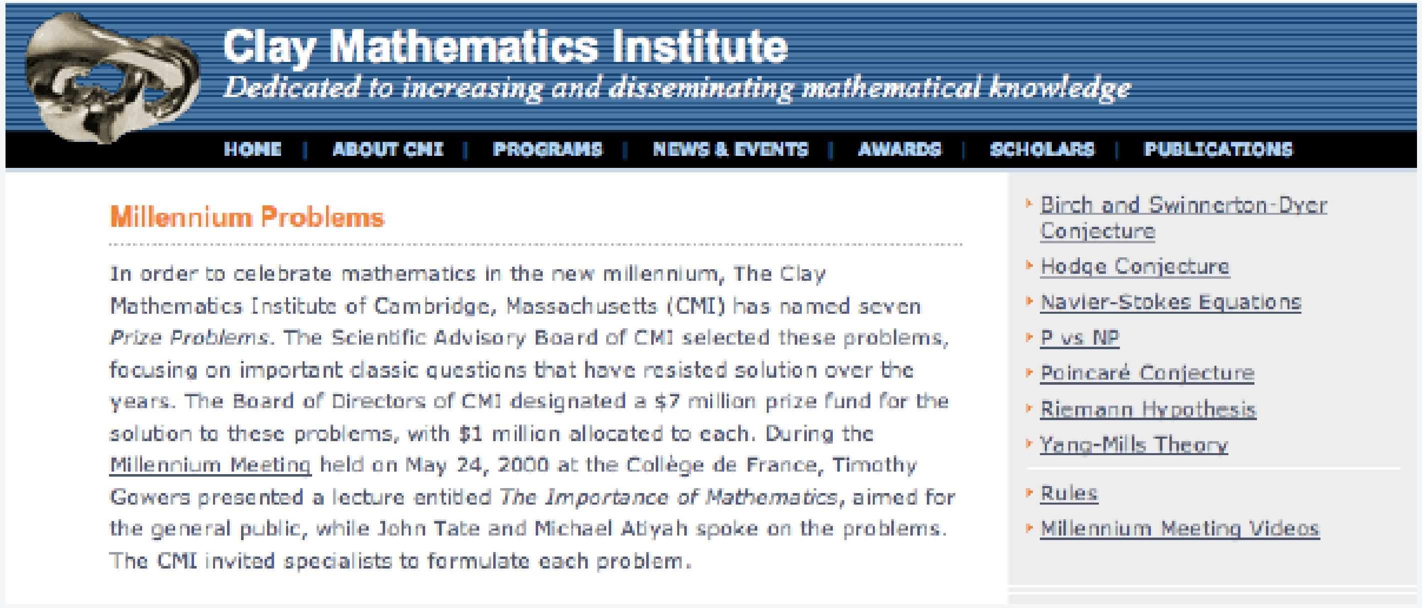
Solusinya adalah: 1 – 2 – 3 – 6 – 7 – 8 – 9 – 10 – 11 – 5 – 4 – 1
→ dapat diverifikasi dengan cepat
→ Menemukan solusinya membutuhkan waktu eksponensial

The $1M question

The Clay Mathematics Institute Millenium Prize Problems:

1. Birch and Swinnerton-Dyer Conjecture
2. Hodge Conjecture
3. Navier-Stokes Equations
4. P vs NP
5. Poincaré Conjecture
6. Riemann Hypothesis
7. Yang-Mills Theory





Millennium Problems

In order to celebrate mathematics in the new millennium, The Clay Mathematics Institute of Cambridge, Massachusetts (CMI) has named seven *Prize Problems*. The Scientific Advisory Board of CMI selected these problems, focusing on important classic questions that have resisted solution over the years. The Board of Directors of CMI designated a $7 million prize fund for the solution to these problems, with $1 million allocated to each. During the Millennium Meeting held on May 24, 2000 at the Collège de France, Timothy Gowers presented a lecture entitled *The Importance of Mathematics*, aimed for the general public, while John Tate and Michael Atiyah spoke on the problems. The CMI invited specialists to formulate each problem.

- Birch and Swinnerton-Dyer Conjecture
- Hodge Conjecture
- Navier-Stokes Equations
- P vs NP
- Poincaré Conjecture
- Riemann Hypothesis
- Yang-Mills Theory
- Rules
- Millennium Meeting Videos

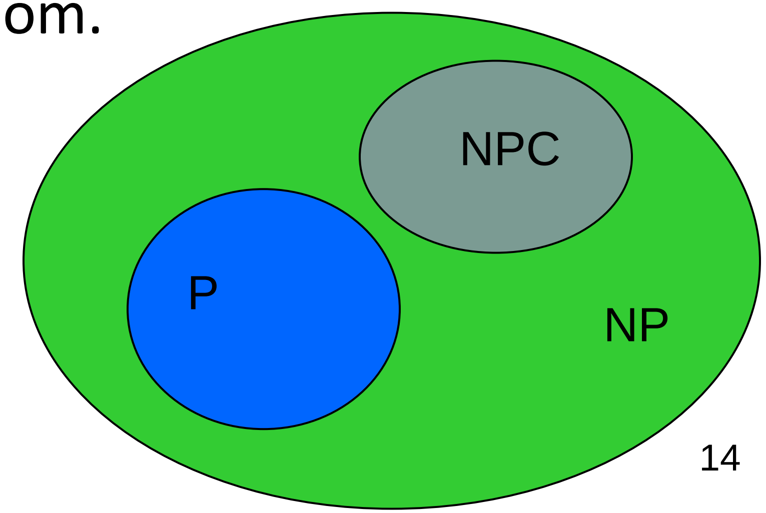
NP-Complete Problems

- NP-Complete (NPC) adalah persoalan NP yang paling menarik perhatian.

- **Definisi NPC.** Sebuah persoalan X dikatakan NPC jika:
  1) X termasuk ke dalam kelas NP
  2) Setiap persoalan di dalam NP dapat direduksi menjadi X dalam waktu polinom

- Konsekuensi dari definisi di atas adalah jika kita dapat menemukan algoritma dengan waktu polinom untuk menyelesaikan persoalan X, maka kita dapat memecahkan semua persoalan di dalam kelas NP dalam waktu polinom.

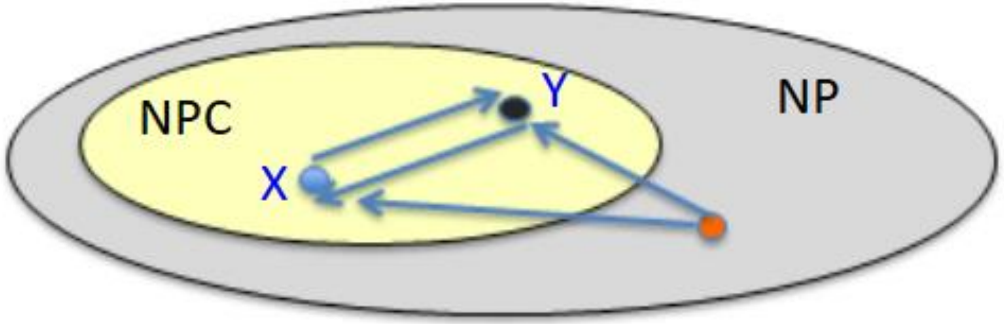
- FYI, nama "NP-Complete" berasal dari:
  - Nondeterministic Polynomial
  - Complete - "Solve one, Solve them all"



Properti NP-Complete

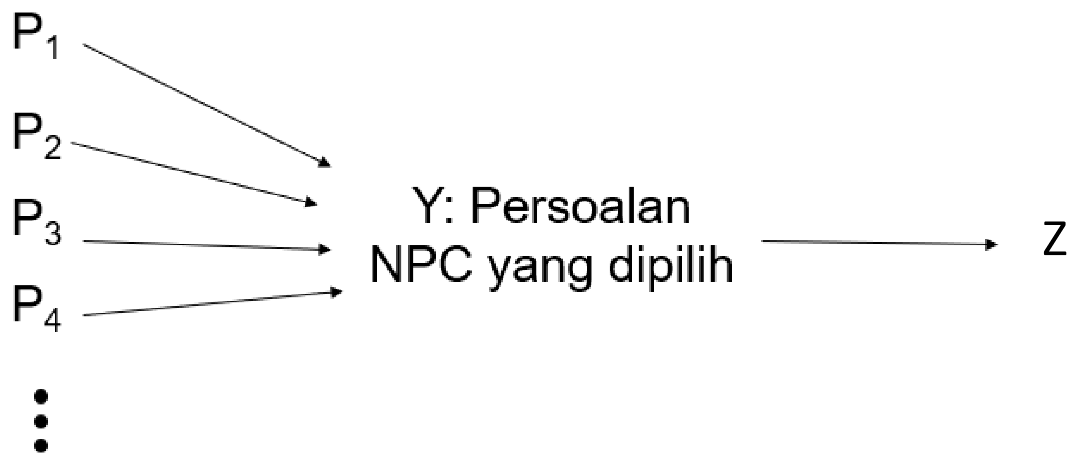
1. Persoalan X adalah NPC jika sembarang persoalan lain di dalam NP dapat direduksi (ditransformasikan) menjadi X dalam waktu polinom

2. Dua persoalan sembarang di dalam NPC dapat ditransformasikan satu sama lain dalam waktu polinom
   - X ditransformasi menjadi Y dalam waktu polinom
   - Y ditransformasi menjadi X dalam waktu polinom



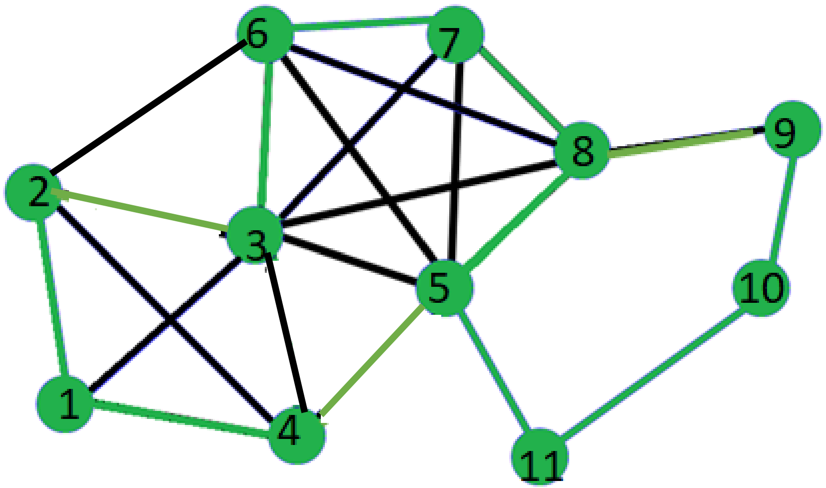
- Cara termudah untuk membuktikan sebuah persoalan baru Z adalah NPC adalah menemukan sebuah algoritma dalam waktu polinom untuk mentransformasikan sebuah persoalan yang sudah dikenal NPC menjadi persoalan Z.

- Jadi, untuk menunjukkan bahwa Z adalah NPC, caranya adalah sebagai berikut:
  1) Tunjukkan bahwa Z adalah anggota NP
  2) Pilih Y yang sesuai dari koleksi persoalan NPC.
  3) Tunjukkan sebuah algoritma dalam waktu polinom untuk mentransformasikan *instance* persoalan Y menjadi *instance* persoalan Z



- **Contoh**: Kita pilih sebuah persoalan dari NPC yaitu sirkuit Hamilton (HCP, *Hamiltonian Circuit Problem*) untuk menunjukkan bahwa persoalan TSDP juga termasuk ke dalam NPC. Jadi, di sini Y = HCP, dan Z = TSDP

Persoalan HCP: Diberikan sebuah graf *G* dengan *n* buah simpul, tentukan apakah graf tersebut mengandung sirkuit Hamilton. Sirkuit Hamilton adalah sirkuit yang melalui setiap simpul di dalam graf *G* dan kembali lagi ke simpul semula.



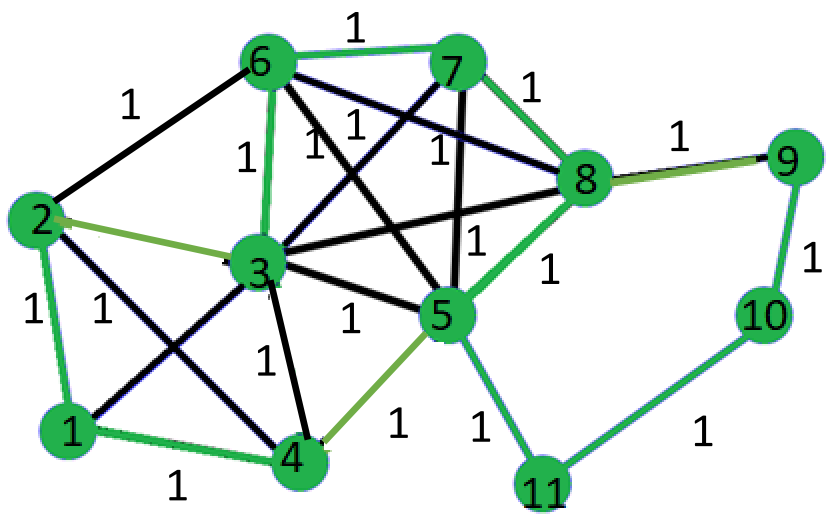
Perhatikan bahwa sirkuit Hamilton di dalam graf G dengan n simpul akan mempunyai n buah sisi

- Untuk mentransformasikan instans HCP menjadi instans TSDP, maka algoritma transformasi yang sederhana adalah sbb:

1. Setiap sisi di dalam graf G diberi nilai (bobot) 1
2. Nyatakan persoalan menjadi TSDP, yaitu adakah tur dengan total bobot $\leq n$?

- Transformasi ini (yaitu memberi bobot setiap sisi dengan nilai 1) membutuhkan waktu polinom, yaitu O(m), m adalah jumlah sisi di dalam graf.

- Dengan transformasi ini, maka persoalan HCP sudah ditransformasi menjadi instans persoalan TSDP.



- Misalkan di dalam graf G = (V, E), |E| = *m*, yaitu jumlah sisi di dalam graf adalah n. Maka, algoritma memberi setiap sisi di dalam graf G dengan 1 adalah sbb:

for tiap sisi $(u, v) \in E$ **do**

    $(u, v) \leftarrow 1$

end

Jumlah pengulangan untuk $(u, v) \leftarrow 1$ adalah sebanyak *m* kali, sehingga:
$T(n) = m = O(m)$ → polinomial

- Transformasi ini memberi sugesti bahwa jika TSDP dapat diselesaikan dalam waktu dalam polinom, maka HCP juga dapat diselesaikan dalam waktu polinom.


You
NP-Complete
Me

You
NP-Complete
Me

- Sejauh ini, lebih dari 300 persoalan yang sudah terbukti NP-complete

- Daftar persoalan yang ternasuk ke dalam NP-Complete:

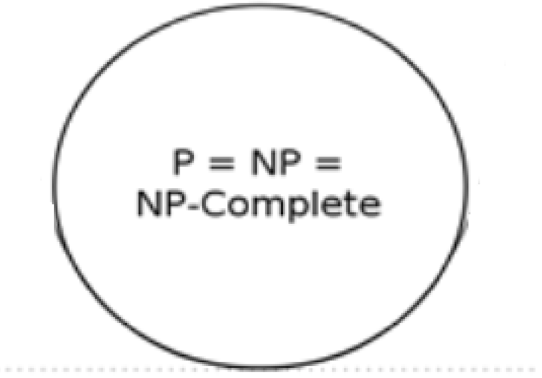
  - SAT (satisfiable problem)
  - TSP (decision problem, atau TSDP)
  - HAMILTONIAN CYCLE problem
  - PARTITION problem
  - SUM OF SUBSET problem
  - CLIQUE problem
  - GRAPH COLORING problem (decision problem)
  - SAT (Satisfiability problem)
  - Vertex cover
  - N-PUZZLE
  - Knapsack problem (decision problem)
  - Subgraph isomorphism problem
  - MINESWEEPER

- Tinjau kembali definisi NPC. Sebuah persoalan X dikatakan NPC jika:

  1. X termasuk ke dalam kelas NP
  2. Setiap persoalan di dalam NP dapat direduksi dalam waktu polinom menjadi X

- Definisi ini menyatakan jika transformasi dari persoalan NP menjadi persoalan NPC dapat dilakukan, maka jika algoritma dalam waktu polinom ditemukan untuk X, maka semua persoalan di dalam NP dapat diselesaikan dengan sangkil.

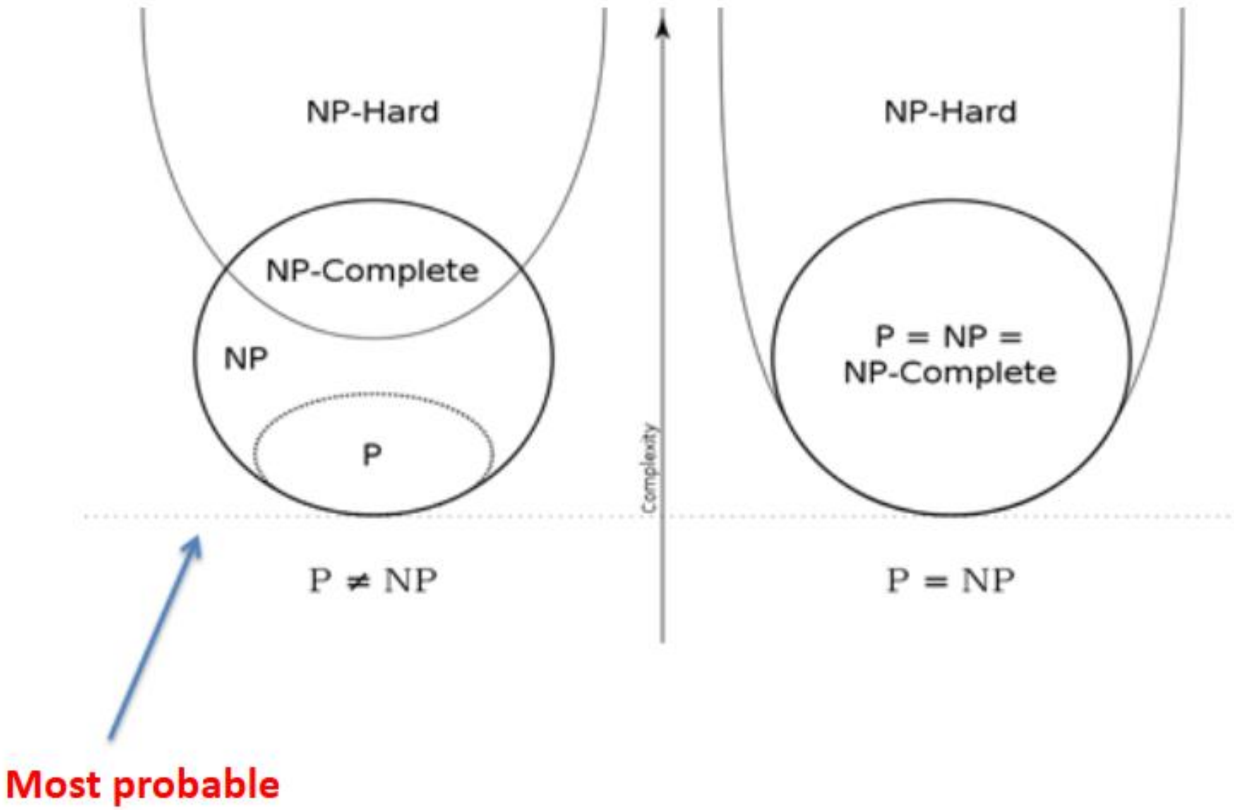
- Dengan kata lain, jika X adalah NPC dan memiliki waktu polinom (yaitu algoritma dalam waktu polinom untuk X ditemukan) maka terjawablah bahwa P = NP.



P = NP

NP-Hard

- NP-hard adalah himpunan persoalan yang sesukar NPC, namun tidak harus berupa persoalan keputusan, bisa persoalan apapun.
- Umumnya memiliki kompleksitas algoritma dengan waktu eksponensial.
- Contoh: TSP (*non decision problem*)



Latihan (diambil dari soal-soal UAS)

1. Diberikan beberapa pernyataan pada gambar berikut terkait P, NP, dan NP-complete. Tentukan apakah setiap pernyataan benar atau salah.

(i) *P* adalah himpunan semua persoalan apapun yang memiliki kebutuhan waktu dalam polinomial

(ii) *NP* adalan himpunan semua persoalan keputusan yang memiliki kebutuhan waktu non-polinomial

(iii) *Halting Problem* tidak termasuk ke dalam kelas NP

(iv) Algoritma non-deterministik selalu memiliki tahap verifikasi dalam waktu polinomial.

(v) Sebuah persoalan *X* dikatakan *NP-complete* jika *X* termasuk ke dalam kelas *NP* dan beberapa persoalan di dalam *NP* lainnya dapat direduksi menjadi instans persoalan X dalam waktu polinomial.

(vi) Jika *A* adalah sebuah persoalan di dalam *NP-complete* dan *B* adalah persoalan *NP* tapi tidak perlu *NP-complete*, maka jika *B* dapat diselesaikan dalam waktu polinomial maka *A* juga dapat diselesaikan dalam waktu polinomial.

(vii) P = NP jika dan hanya jika persoalan di dalam *NP-complete* dapat diselesaikan dalam waktu polinomial.

2. Misalkan persoalan A adalah NP-complete dan persoalan B adalah NP tapi tidak perlu NP-complete. Manakah dari pernyataan-pernyatan berikut yang benar? Jawaban bisa lebih dari satu. Jelaskan alasannya.

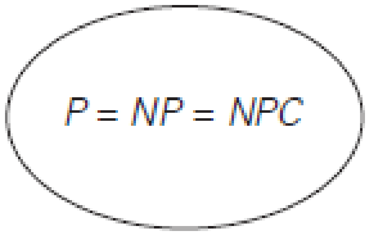
I. Algoritma dengan waktu polinom untuk A mengimplikasikan bahwa P = NP.

II. Algoritma dengan waktu polinom untuk B mengimplikasikan bahwa P = NP.

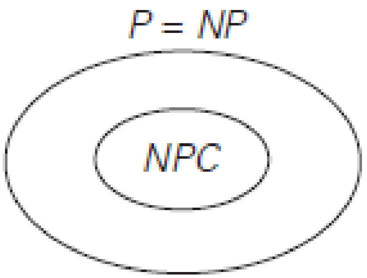
III. Algoritma dengan waktu polinom untuk A mengimplikasikan algoritma dengan waktu polinom untuk B.

3. Which of the following diagrams do not contradict the current state of our knowledge about the complexity classes P, NP, and NPC (NP-complete problems)?

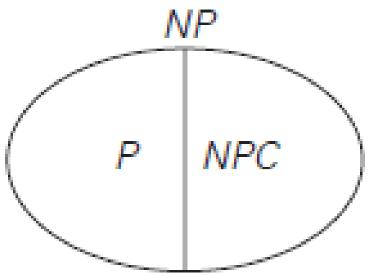
a.



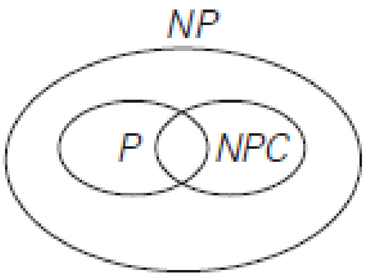
b.



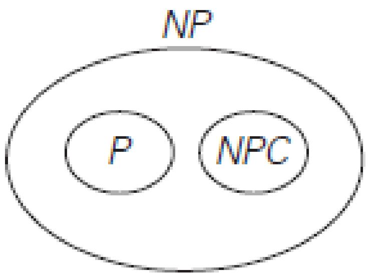
c.



d.



e.



(Sumber:Levitin)

T A M A T