Metoda Pembuktian

Pembuktian Teorema

Pembuktian Langsung:

Suatu implikasi ***p*→*q*** dapat dibuktikan dengan menunjukkan bahwa jika ***p* benar, maka *q* juga benar**.

Contoh: Berikan pembuktian langsung dari teorema "Jika *n* ganjil, maka n^2 juga ganjil."

Ide: Asumsikan bahwa hipotesis dari implikasi ini benar (*n* ganjil). Lalu gunakan aturan inferensi dan teorema yang telah diketahui untuk menunjukkan bahwa *q* juga benar (yaitu, n^2 juga ganjil).

Pembuktian Teorema

***n* ganjil**, maka *n* bisa dituliskan sebagai ***n* = 2*k* + 1**, dimana k bilangan bulat.

Akibatnya,

$$\begin{aligned} n^2 &= (2k + 1)^2. \\ &= 4k^2 + 4k + 1 \\ &= 2(2k^2 + 2k) + 1 \\ &= 2m+1, \text{ dimana } m= (2k^2 + 2k) \end{aligned}$$

Karena n^2 dapat dituliskan sebagai 2m+1, untuk bilangan bulat m, maka n^2 adalah bilangan ganjil.

=> Terbukti !

Pembuktian Teorema

Pembuktian tak langsung: “ $\neg q \rightarrow \neg p$ “
Suatu implikasi **$p \rightarrow q$** ekivalen dengan bentuk **kontra-positif** nya, **$\neg q$** $\rightarrow$ **$\neg p$** (bisa ditunjukkan dengan tabel logika). Oleh karena itu, kita dapat membuktikan $p \rightarrow q$ dengan menunjukkan bahwa, jika q salah, maka p juga salah.

Contoh: Berikan bukti tak langsung teorema
“Jika 3n + 2 ganjil, maka n adalah ganjil.”

Ide: Asumsikan bahwa kesimpulan dari implikasi ini salah (n genap). Kemudian gunakan aturan inferensi dan teorema yg telah diketahui untuk menunjukkan bahwa p juga salah (3n + 2 genap).

> “jika n **tidak ganjil**, maka 3n+2 juga **tidak ganjil**”. Bilangan tidak ganjil adalah bilangan genap, oleh karena itu, pernyataan tsb ekivalen dng ”jika n genap, maka 3n+2 juga genap.”. Ingat, bil. genap dpt dituliskan 2m, untuk m bulat.

Pembuktian Teorema

n genap ….. (negasi dari *q* ganjil)

Maka *n* bisa dinyatakan sbg $n = 2k$, dimana k bulat.

Dengan demikian $3n + 2 = 3(2k) + 2$

$$= 6k + 2$$

$$= 2(3k + 1)$$

=> Terbukti

Oleh karena itu, 3n + 2 genap (*p* salah, jadi sudah ditunjukkan $\neg q \rightarrow \neg p$).

Kita telah menunjukkan bahwa kontrapositif dari implikasi bernilai benar,jadi implikasinya sendiri juga benar (Jika 2n + 3 ganjil, maka n ganjil).

(Prinsip) Induksi

Induksi matematika adalah alat yang berguna untuk membuktikan bahwa predikat tertentu bernilai benar (berlaku) untuk (semua) **bilangan cacah**.

Prinsip ini tidak dapat dipakai untuk menemukan suatu teorema, tapi hanya untuk melakukan pembuktian.

Induksi

Jika kita punya fungsi proposisi P(*n*), dan kita ingin membuktikan bahwa P(*n*) benar untuk sebarang bilangan cacah *n*, maka kita melakukan langkah-langkah berikut ini:

<u>(1) Langkah Dasar:</u> Menunjukkan bahwa P(1) [atau P(0) jika n non-negatif] benar

<u>(2) Langkah Induktif:</u> Menunjukkan bahwa jika P(n) (berlaku) maka P(n+1) , yakni (P(n)$\rightarrow$P(n+1) juga berlaku) untuk sebarang n$\in$N.

<u>(3) Kesimpulan:</u> Maka P(n) harus benar untuk sebarang n$\in$N.

Efek domino dari induksi

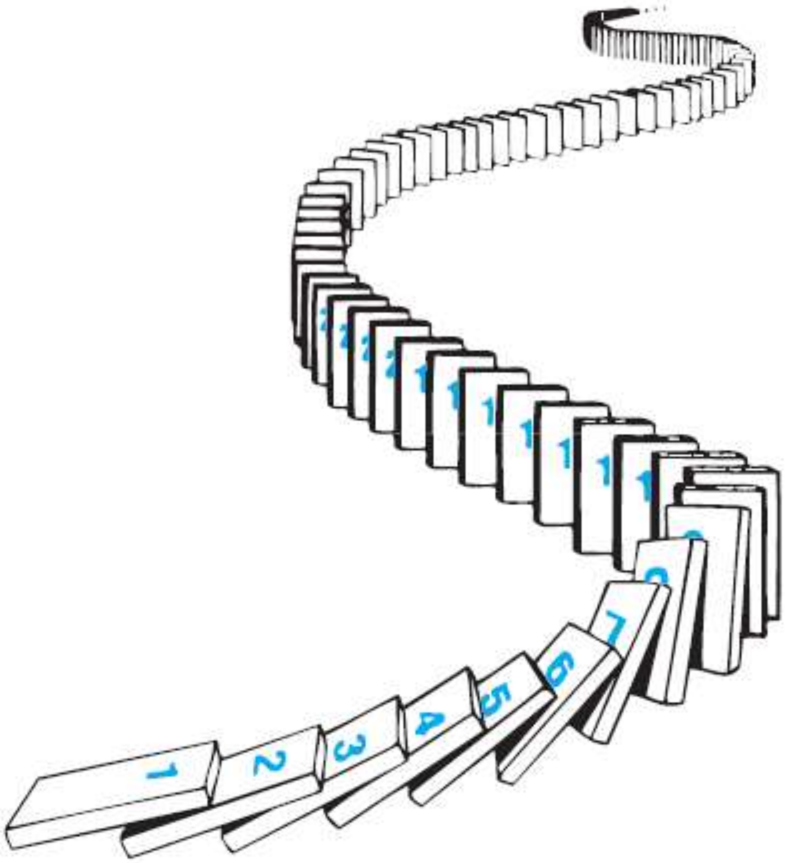


FIGURE 2 **Illustrating How Mathematical Induction Works Using Dominoes.**

Induksi

Contoh:

Tunjukkan bahwa $n < 2^n$ untuk semua bilangan bulat positif n. (n=1, 2, ….)

Misalkan P(n) adalah proposisi “$n < 2^n$.”

(1) **Langkah Dasar**: kita tunjukkan bahwa P(1) benar.

P(1) benar, karena $1 < 2^1$ (= 2).

Induksi

(2) Langkah Induktif: tunjukkan bahwa, jika P(n) benar, maka P(n +1) juga benar atau P(n)$\rightarrow$P(n+1) berlaku.

Asumsikan bahwa $n < 2^n$ benar.

Kita perlu menunjukkan bahwa P(n + 1) adalah benar, yaitu $n + 1 < 2^{n+1}$

Dimulai dari $n < 2^n$, tambahkan kedua ruas dengan 1, dan substitusikan (n+1):

$n + 1 < 2^n + 1 \leq 2^n + 2^n = 2^{n+1}$

Oleh karena itu, jika $n < 2^n$ maka $n + 1 < 2^{n+1}$

Induksi

(3) Kesimpulan: Maka P(n) haruslah benar untuk sebarang bilangan bulat positif.

$n < 2^n$ adalah benar untuk sebarang bilangan bulat positif.

Akhir dari pembuktian.

Induksi

Contoh lain (formula "Gauss"):

$$1 + 2 + \ldots + n = n\,(n + 1)/2$$

1. Tunjukkan bahwa P(1) adalah benar. (langkah dasar)

Untuk n = 1 kita peroleh 1 = 1. Benar.

Induksi

2. Tunjukkan bahwa jika $P(n)$ maka $P(n+1)$ untuk sebarang $n \in N$. (langkah induktif);

$1 + 2 + \ldots + n = n\,(n + 1)/2$

tambahkan satu suku berikutnya, $(n+1)$

$$1 + 2 + \ldots + n + (n + 1) = n\,(n + 1)/2 + (n + 1)$$
$$= (2n + 2 + n\,(n + 1))/2$$
$$= (2n + 2 + n^2 + n)/2$$
$$= (2 + 3n + n^2)/2$$
$$= (n + 1)\,(n + 2)/2$$
$$= (n + 1)\,((n + 1) + 1)/2$$

Induksi

3. Maka P(n) haruslah benar untuk sebarang $n \in N$. (kesimpulan)

Jadi

$1 + 2 + \ldots + n = n(n + 1)/2$ adalah benar untuk semua $n \in N$.

Akhir dari pembuktian.

Induksi

Ada teknik pembuktian lain yang sangat mirip dengan prinsip induksi matematika.

Prinsip ini disebut sebagai **prinsip kedua dari induksi matematika.**

Prinsip ini dapat dipergunakan untuk membuktikan bahwa suatu fungsi proposisi P(n) bernilai benar untuk sebarang bilangan cacah n.

Rekursi

Definisi Rekursif

Rekursif adalah suatu prinsip yang sangat dekat hubungannya dengan induksi matematika.

Didalam **definisi** rekursif, suatu objek didefinisikan dari dirinya sendiri.

Kita dapat mendefinisikan **deretan**, **fungsi** dan **himpunan** secara rekursif.

Deretan yang Didefinisikan Secara Rekursif

Contoh:

Suatu deretan $\{a_n\}$ dinyatakan sebagai

$$a_n = 2^n \quad \text{untuk } n = 0, 1, 2, \ldots .$$

Deretan tsb dapat juga didefinisikan secara **rekursif**:

$$a_0 = 1$$
$$a_{n+1} = 2a_n \quad \text{untuk } n = 0, 1, 2, \ldots$$

Jelas bahwa induksi dan rekursi adalah prinsip yang sama.

Deretan yang Didefinisikan Secara Rekursif

Kita dapat menggunakan metoda berikut ini untuk mendefinisikan suatu fungsi dengan domain bilangan cacah:

1. Tentukan nilai fungsi pada (argumen) nol.
2. Berikan aturan untuk mencari nilainya pada sebarang bilangan bulat berdasarkan dari nilainya pada bilangan bulat yang lebih kecil.

Yang demikian ini disebut **definisi rekursif** atau **induktif** dari suatu fungsi.

Deretan yang Didefinisikan Secara Rekursif

Contoh:

$f(0) = 3$

$f(n + 1) = 2f(n) + 3,$

Maka

$f(0) = 3$

$f(1) = 2f(0) + 3 = 2 \cdot 3 + 3 = 9$

$f(2) = 2f(1) + 3 = 2 \cdot 9 + 3 = 21$

$f(3) = 2f(2) + 3 = 2 \cdot 21 + 3 = 45$

$f(4) = 2f(3) + 3 = 2 \cdot 45 + 3 = 93$

Deretan yang Didefinisikan Secara Rekursif

Bagaimana mendefinisikan fungsi faktorial $f(n) = n!$ secara rekursif?

$$f(0) = 1$$
$$f(n + 1) = (n + 1)f(n)$$

$$f(0) = 1$$
$$f(1) = 1f(0) = 1\cdot 1 = 1$$
$$f(2) = 2f(1) = 2\cdot 1 = 2$$
$$f(3) = 3f(2) = 3\cdot 2 = 6$$
$$f(4) = 4f(3) = 4\cdot 6 = 24$$

Deretan yang Didefinisikan Secara Rekursif

Suatu contoh yang terkenal adalah Bilangan Fibonacci:

$$f(0) = 0, f(1) = 1$$

$$f(n) = f(n - 1) + f(n - 2)$$

Maka,

$$f(0) = 0$$

$$f(1) = 1$$

$$f(2) = f(1) + f(0) = 1 + 0 = 1$$

$$f(3) = f(2) + f(1) = 1 + 1 = 2$$

$$f(4) = f(3) + f(2) = 2 + 1 = 3$$

$$f(5) = f(4) + f(3) = 3 + 2 = 5$$

$$f(6) = f(5) + f(4) = 5 + 3 = 8$$

Himpunan yang Didefinisikan Secara Rekursif

Untuk mendefinisikan suatu himpunan secara rekursif, kita memerlukan dua hal berikut:

1. Nilai awal (sekumpulan **elemen-elemen awal**)
2. Aturan (untuk mendefinisikan elemen-elemen **tambahan** dari elemen yang sudah ada didalam himpunan).

Contoh: Mis. S didefinisikan secara rekursif sbb:

$3 \in S$: nilai awal

$(x + y) \in S$ jika $(x \in S)$ dan $(y \in S)$: aturan

S adalah himpunan dari bilangan bulat positif yang dapat dibagi 3.

Himpunan yang Didefinisikan Secara Rekursif

Bukti:

Misalkan A himpunan semua bilangan bulat positif yang dapat dibagi 3.

Untuk menunjukkan A = S, kita harus membuktikan bahwa

$$A \subseteq S \text{ dan } S \subseteq A.$$

Bag. I: Untuk membuktikan bahwa $A \subseteq S$, kita harus menunjukkan bahwa setiap bilangan bulat positif yang dapat dibagi 3 terkandung didalam S.

Untuk menunjukkan hal ini kita akan memakai induksi matematika.

Himpunan yang Didefinisikan Secara Rekursif

Mis. P(n) adalah pernyataan "3n anggota S".

Langkah dasar: P(1) benar, karena $3 \in S$.

Langkah induktif: Untuk menunjukkan:
Jika P(n) benar, maka P(n + 1) benar.

Asumsikan 3n anggota S. Karena 3n anggota S dan 3 adalah anggota S, dari definisi rekursif S maka $3n + 3 = 3(n + 1)$ adalah juga anggota S.

Kesimpulan Bagian I: $A \subseteq S$.

Himpunan yang Didefinisikan Secara Rekursif

Bag II: Akan ditunjukkan: $S \subseteq A$.

Langkah dasar: Akan ditunjukkan:
Semua elemen awal dari S adalah juga anggota A.

3 anggota A. Benar.

Langkah induktif: Akan ditunjukkan:
$(x + y)$ adl anggota A manakala x dan y anggota S.

Jika x dan y keduanya anggota A, maka $3|x$ dan $3|y$.
Seperti telah kita ketahui, ini berarti $3|(x + y)$.

Kesimpulan Bagian II: $S \subseteq A$.

Kesimpulan Keseluruhan: $A = S$.

Himpunan yang Didefinisikan Secara Rekursif

Contoh lain:

Formula yang *well-formed* dari variabel, bilangan dan operator {+, -, *, /, ^} didefinisikan sbb:

> x adalah suatu formula yang *well-formed* jika x merupakan suatu bilangan atau variabel.
>
> $(f + g)$, $(f - g)$, $(f * g)$, (f / g), $(f \wedge g)$ adalah formula yang *well-formed* jika f dan g formula yang *well-formed*.

Himpunan yang Didefinisikan Secara Rekursif

Dengan definisi ini, kita dapat membentuk formula-formula, misalnya:

$(x - y)$

$((z / 3) - y)$

$((z / 3) - (6 + 5))$

$((z / (2 * 4)) - (6 + 5))$

Algoritma rekursif

Algoritma Rekursif

Suatu algoritma disebut **rekursif** jika algoritma tsb memecahkan suatu permasalahan dengan cara mereduksi permasalahan itu menjadi permasalahan sama dengan masukan yang lebih kecil.

Contoh I: Algoritma Euklidian Rekursif

```
procedure gcd(a,b: nonnegative integers dng a<b)
      if a=0 then gcd(a,b):=b
      else gcd(a,b) := gcd(b mod a,a)
```

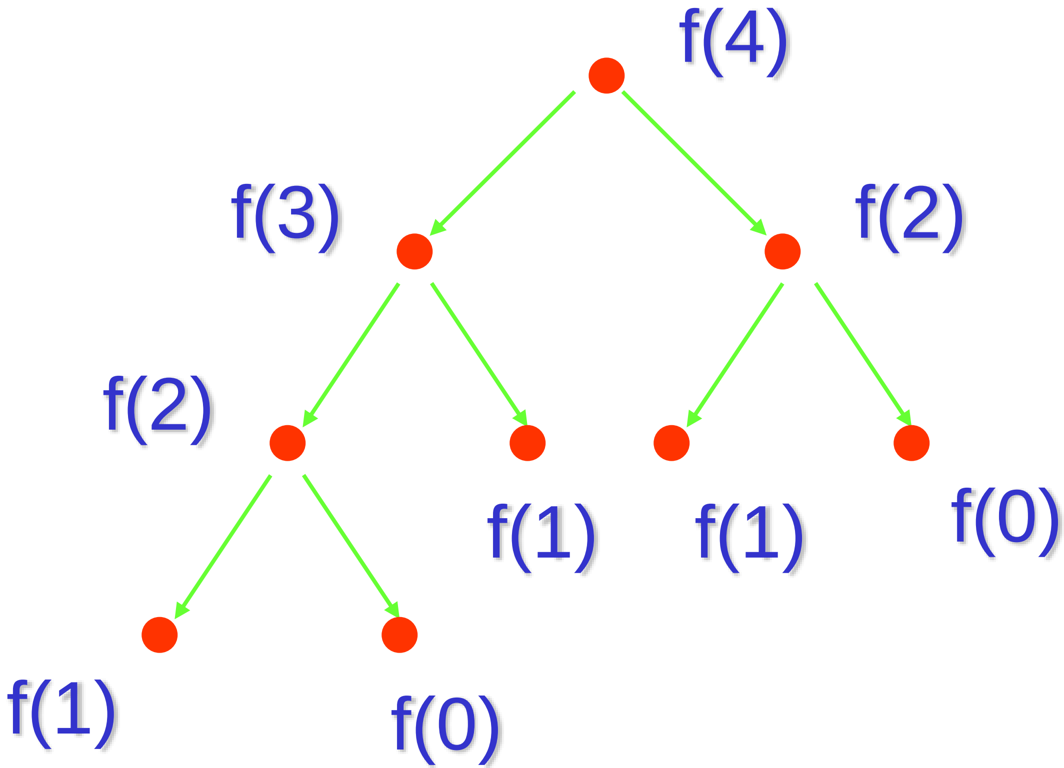
Algoritma Rekursif

Contoh II: Algoritma Fibonacci Rekursif

```
procedure fibo(n: nonnegative integer)
  if n = 0 then fibo(0):= 0
    else if n = 1 then fibo(1):= 1
      else fibo(n):= fibo(n-1)+fibo(n-2)
```

Algoritma Rekursif

Evaluasi algoritma Fibonacci rekursif:



Algoritma Rekursif

```
procedure iterative_fibo(n: nonnegative integer)
if n = 0 then y := 0
else
begin
      x := 0
      y := 1
      for i := 1 to n-1
      begin
            z := x + y
            x : = y
            y := z
      end
end {y adalah bilangan Fibonacci ke-n}
```

Algoritma Rekursif

Untuk setiap algoritma rekursif, terdapat algoritma iteratif yang **ekivalen**.

Algoritma rekursif sering kali **lebih singkat, lebih elegan,** dan **lebih mudah dipahami** dibandingkan algoritma iteratif-nya.

Tetapi, algoritma iteratif biasanya **lebih efisien** menurut ukuran kompleksitas ruang dan waktu.

Contoh populer: L-System

L: Lindenmayer

- D0L-System, bentuk paling sederhana dari L-system
- Example:
  - Alphabet = {a,b}
  - Rules = {a → ab, b → a}
  - Axiom: b

Syntax of a production rule:

Initiator → Generator

```
      b
      |
      a
      └
     a b
    ┘  |
   a b  a
  ┘  |  └
  a b a a b
_/ / ┘ └ \
a b a a b a b a
```

Example of a derivation in a DOL-System

Segitiga Sierpinski

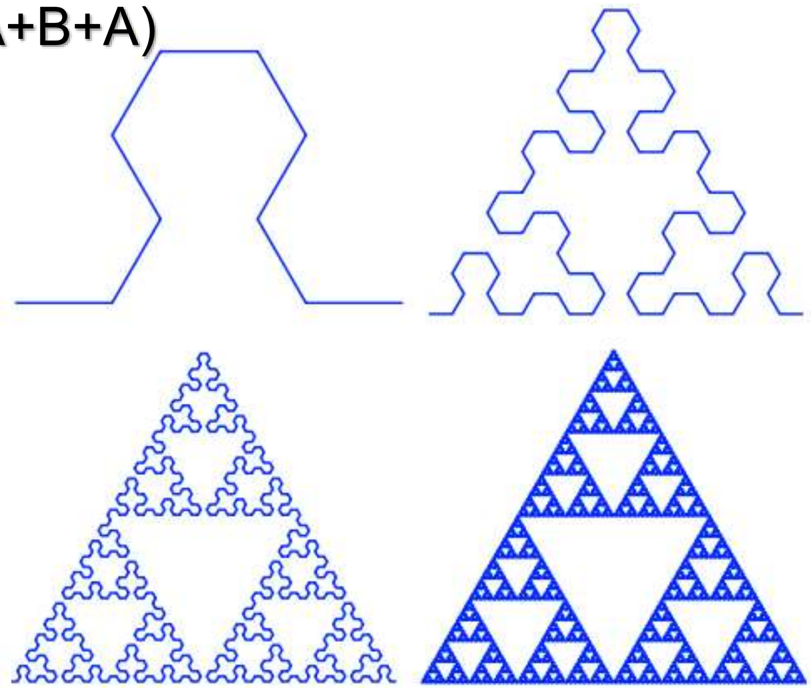
variables : A B

constants : + −

start : A

rules : (A → B−A−B), (B → A+B+A)

angle : 60°



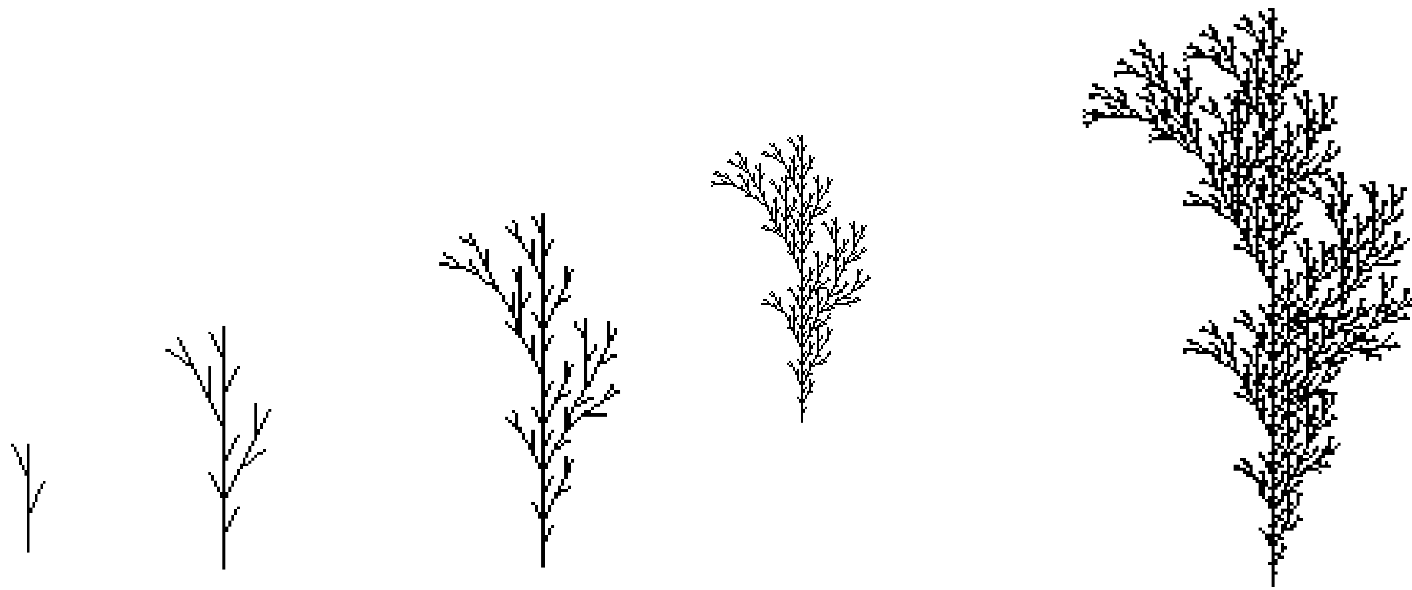
Evolution for n = 2, n = 4, n = 6, n = 8

Tanaman fraktal

w: F

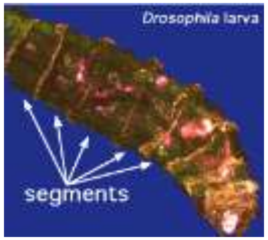
p: F → F[-F]F[+F][F]

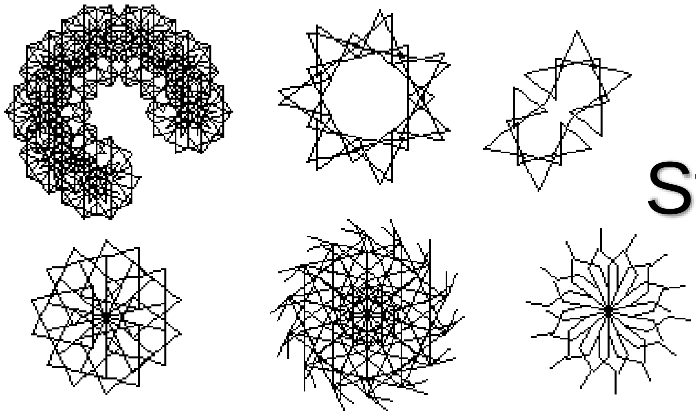
Angle (δ) = 60º



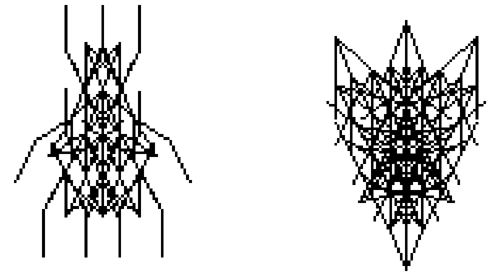
n = 1 - 5

Binatang fraktal ... ?

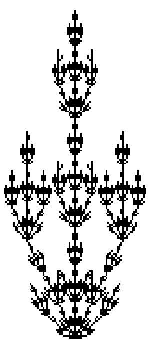




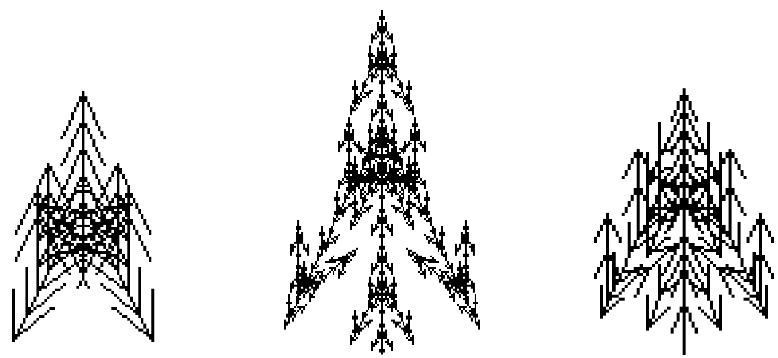
Stars



Animals



Candle stick



Rockets

SELESAI