Kompleksitas

→ Jumlah operasi eksekusi

O(...)

- Linear O(1) → 1 detik
  (1) Rumus Matematika
  (2) If - else biasa
  (3) Tidak ada loop / Rekursi

$$\frac{N(N+1)}{2}$$

→ operasi matematika

- Non - Linear
  - → Iterasi : for / while

For i=1 to N :

1
2
3
⋮
N

} O(N)

While i = 1 to N → O(N)

for i = 1 to ... → O(...)

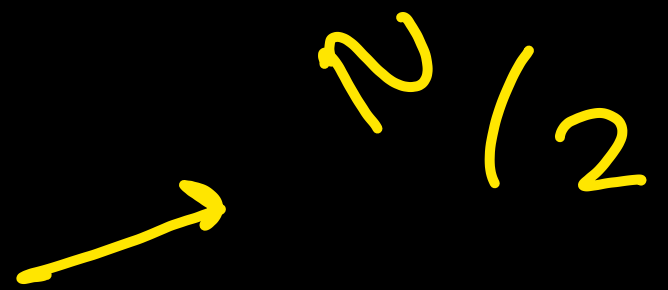
```
for int i = 1; i<=10; i+=2;
```

1
3
5
7
9

Mencerak bi
ganjil dri
1 - 10

5 → jml iterasi

for int i = 1; i<=N; i+=2 ->
1
3
5
...
Bil ganjil <=N



Banyak operasi -> Banyak bil ganjil dari 1 -- N

Kompleksitas = $O(N/2)$

For loop saling Lepas

```
for int i = 1; i<=N; i++{
}

for int j = 1; j<=M; j++{

}
```

O(N)

O(M)

Kompleksitas : O(N+M)

```
for int i = 1; i<=N; i++{
}
```

O(N)

Kompleksitas

$O(N + \frac{M}{3})$

```
for int j = 1; j<=M; j+=3{

}
```

O(M/3)

1, 4, 7, ... , $j + 3 \leq M$

M = 10 → j: 1, 4, 7, 10

M = 12 → 1, 4, 7, 10

$12/3 = \lceil 4 \rceil = 4$

M = 20 → 7 → 1, 4, 7, 10, 13, 16, 19

$\frac{20}{3} = \lceil 6, \ldots 7 \rceil = 7$

$O(N)$ - $N = 10^8$

Prosessor komp. moderne

→ 1 sekk

Rules of thumb

$O(N)$ - $N = 10^{10}$

10^8 → 1 sekk

10^{10} ≈ 1 sekk

dyn kompleksitet $O(N)$

1 <= N <= 10^6 , O(N) Waktu program <= 1 detik ga?

$10^8 \rightarrow$ 1 detik

$10^6 < 10^8 \rightarrow$ < 1 detik

1 <= N <= 10^12 , O(N) Waktu program <= 1 detik ga?

$10^{12} > 10^8 \rightarrow$ > 1 detik

1<= N <= 10^12 Kompleksitas O(N/10^9) waktu program <= 1 detik?

$N = 10^{12} \rightarrow O\left(\frac{N}{10^9}\right)$

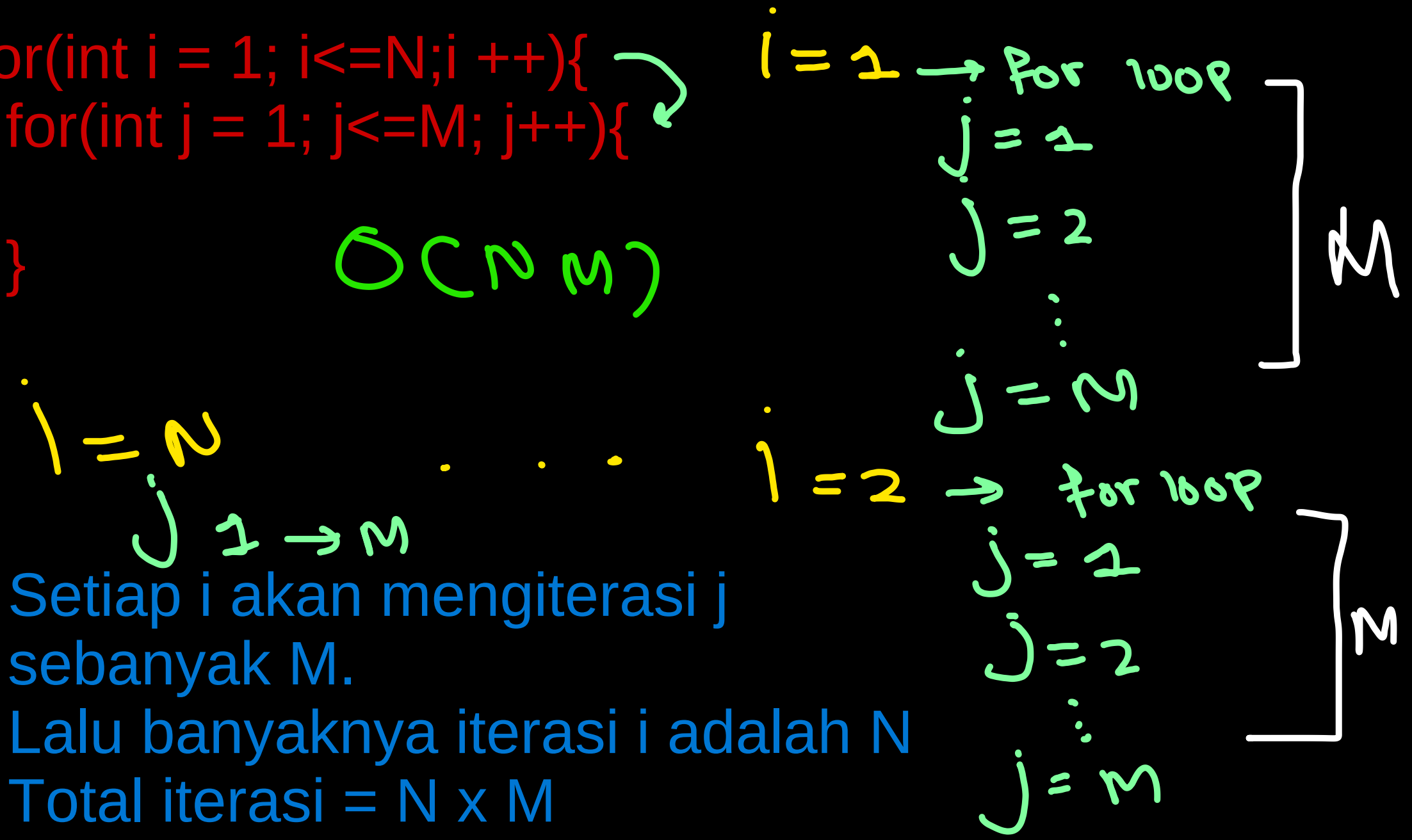
$= \frac{10^{12}}{10^9} = 10^3$ (<= 1 detik)

→ Nested

C for dalam for / while - while)

```
for(int i = 1; i<=N;i ++){
  for(int j = 1; j<=M; j++){

  }
}
```



Setiap i akan mengiterasi j sebanyak M.
Lalu banyaknya iterasi i adalah N
Total iterasi = N x M

1<= N <= 10^3

1 <= M <= 10^7

bisa gak <= 1 detik kompleksitasnya O(NM)?

O(NM) → $10^3 \cdot 10^7 = 10^{10} > 10^8$ (> 1 detik)

gbs

```
for(int i = 1; i<=N + 100;i ++){
  for(int j = 1; j<=M; j++){
     for(k = 1; k<=L; k++{
      for ....
   }
   }
}
```

O(NML ...)

```
for(int i = 1; i<=N; i++){
for(int i = 1; i<=N; i++){
}
}
```

→ $O(N^2)$

→ logaritmik

($\log = 2\log$)

$\log N = 2\log N$

$\neq 1\log N$ salah

```
for(int  i =N; i>=1; i /= 2){


}
```

N = 100
i = 100
⋮
i = 1

} i dibagi 2 terus

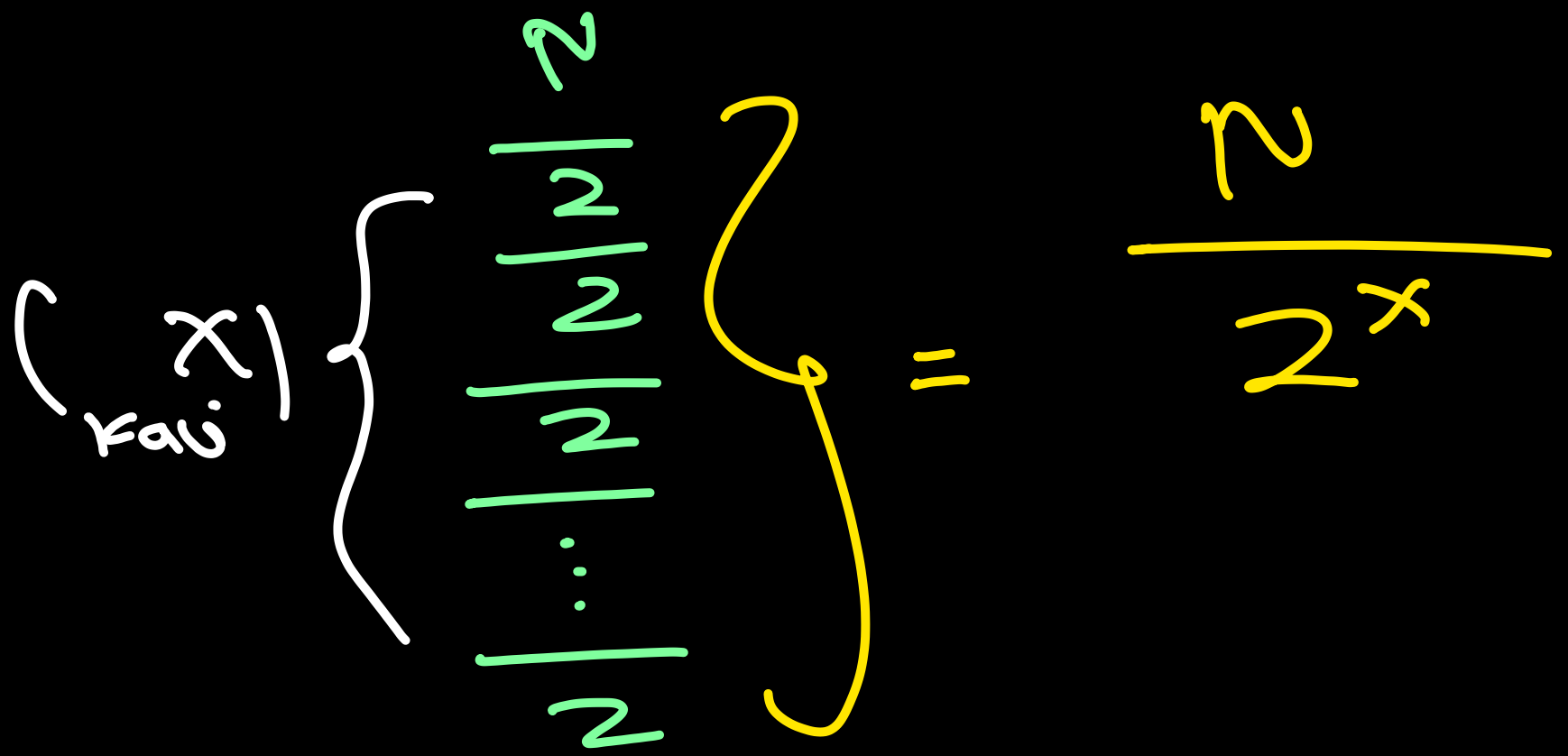
N = 10
i = 10
i = 5
i = 2
i = 1

} 4x

Total iterasi → N s/d 1

x = banyak iterasi

$$\frac{\frac{N}{2}}{2} = \frac{N}{2 \times 2}, \quad \frac{\frac{\frac{N}{2}}{2}}{2} = \frac{N}{2 \times 2 \times 2}$$



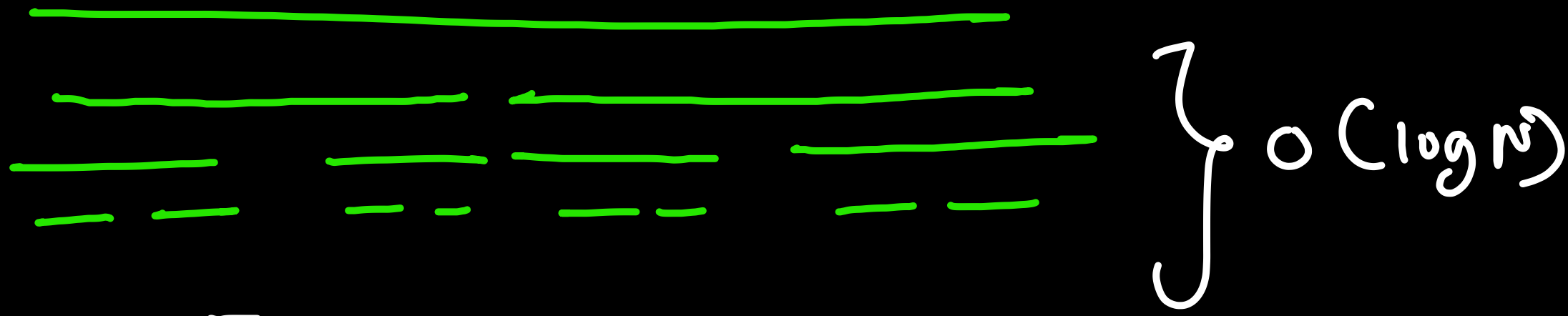
$$= \frac{n}{2^x}$$

$$\frac{n}{2^x} = 1 \rightarrow n = 2^x$$

$$x = 2 \log n$$

$$x = \log_2 N$$

Binary search / Merge sort



$O(N^2)$

$N = 10 \rightarrow t(O(N)) \rightarrow t(100)$

$t(100) = 1$ detik

$t(10^2) = 1$ detik

$N = 100? \rightarrow t(100^2) = t(10^4)$

$10^2 \rightarrow 1$ detik

$10^4 \rightarrow$

Perbandingan senilai

$$\frac{10^9}{10^2} = \frac{x}{1}$$

$10^9 = 10^2 x$

$x = 10^2$ detik

$= 100$ detik

```
for (int i = 0; i < N; i++) {
    if (i * i > N) {
        break;
    }
}
```

} iterasi berhenti saat

$i \times i > N$

$N < i^2$ } $i^2 = N$

$i = \sqrt{N}$

$O(\sqrt{n})$

```
for (int i = 0; i < N; i++) {
    counter = i;
    while (counter > 0) {
        counter = counter / 2;
    }
    counter = 0;
    while (counter * counter < N) {
        counter = counter + 1;
    }
}
```

O(N)

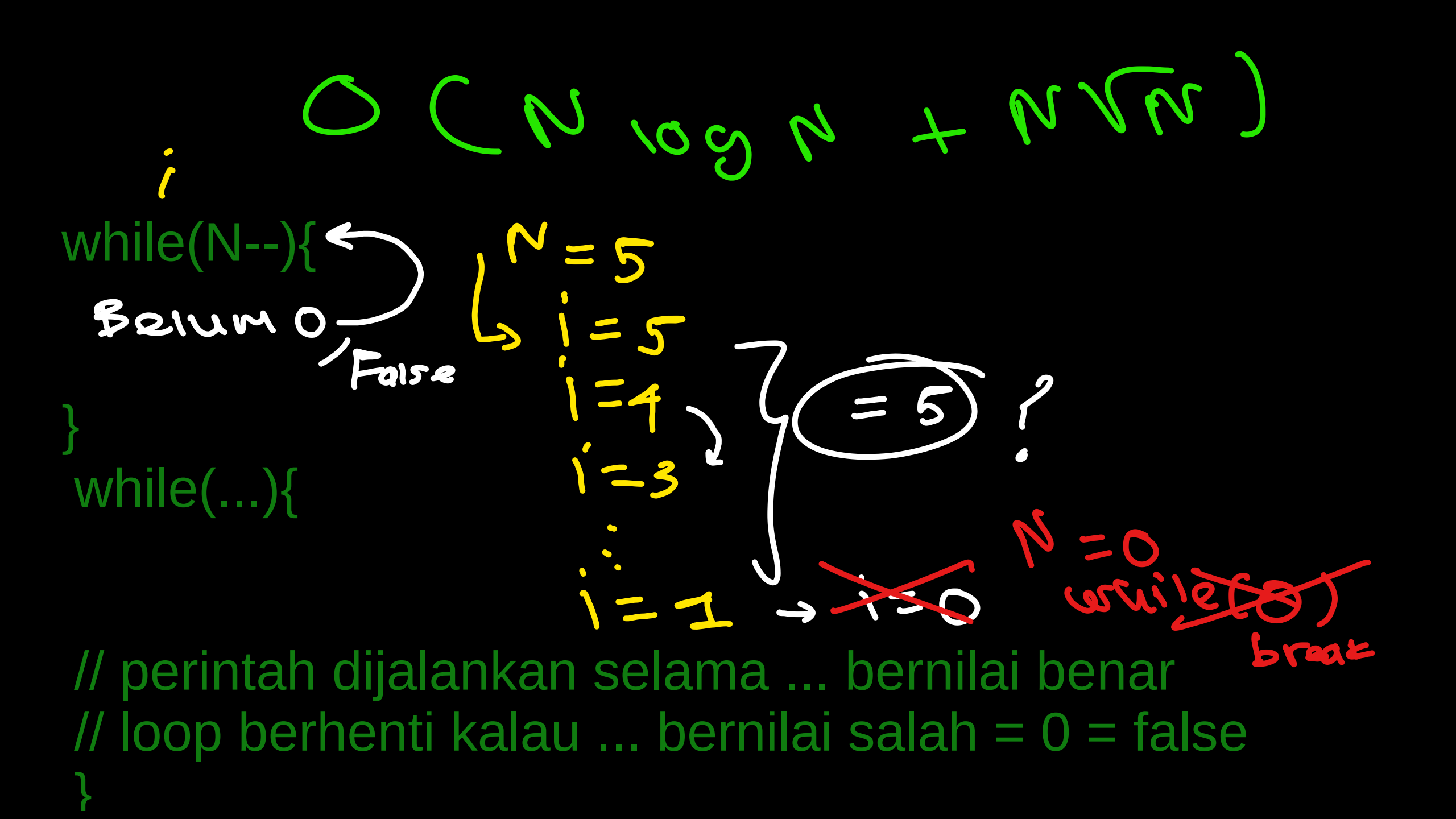
O(log N)

O(√N)

O(log N + √N)

O(N (log N + √N))

saving leaps


O(N log N + N√N)
i
while(N--){
Belum 0
False
}
while(...){
N = 5
i = 5
i = 4
i = 3
i = 1
i = 0
= 5 ?
N = 0
while(0)
break
// perintah dijalankan selama ... bernilai benar
// loop berhenti kalau ... bernilai salah = 0 = false
}

apakah N prima ?

* Cara Naif : 1 - bil

ada Faktor >2 ?

OC N bil

beapa yg habis membagi bil

Faktor

* Sieve E ...

Cek apakah i*i habis membagi bil

16 → {1, 2, 4, 8, 16}

16 habis dibagi 2

Maka 16 habis dibagi $2^2 = 4$

$O(N\sqrt{bil})$ →

$10^3 \cdot \sqrt{10^6} = 10^3 \cdot 10^3 = 10^6 < 10^8$

< 1 detik

★ Solusi Naif

T = 5

1 → fakt(1) = 1 × fakt(0) = 1 × 1

2 → fakt(2) = 2 × fakt(1) = 2 × 1 × fakt(0) = 2

3 → fakt(3) → fakt(2) → fakt(1)

4 → fakt(4) → fakt(3) → fakt(2) → fakt(1)

5 → fakt(5) → ... fakt(1)

fakt(N) : O(N)

Kasus uji sebanyak T, setiap kasujinya

O(N) : O(TN)

$10^7 \cdot 10^3 > 10^9 \rightarrow$ > 1 detik

$T = 5$

1 → fakt(1) = 1

2 → fakt(2) = 2 x fakt(1) = 2

3 → fakt(3) = 3 x fakt(2) = 3x2 = 6

4 → fakt(4) = 4 x fakt(3) = 4 x 6

5 → fakt(5) = 5 x fakt(4) = 5 x 24

Memoisasi

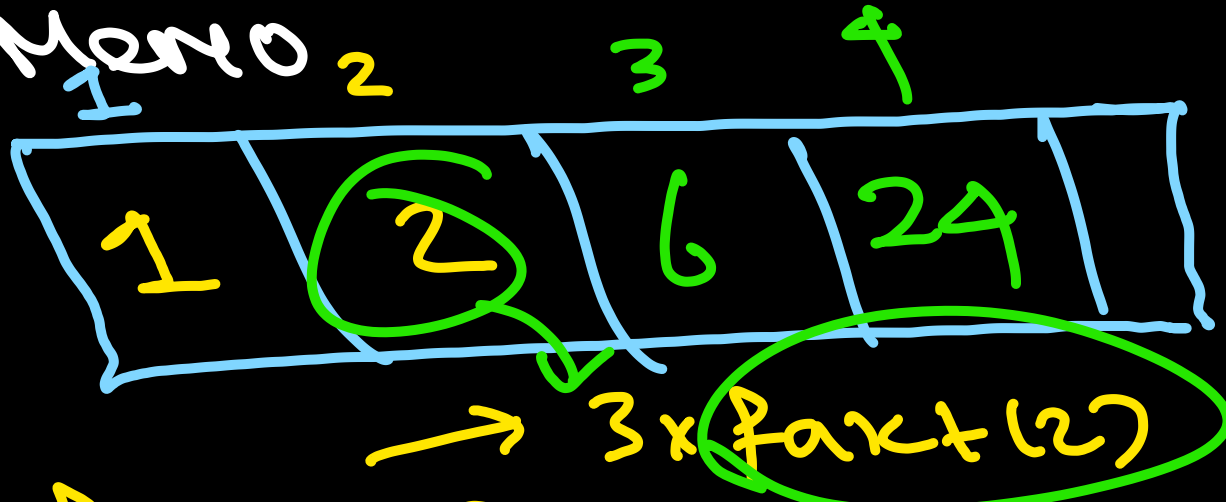
O(N)

- Kita hitung Faktorial (N), kita ingat hasil perhitungan kita tadi
- Jadi semisal ada yang perlu kita ambil kita tinggal ambil

fakt(1) = 1

fakt(2) = 2

fakt(4) = 4 x fakt(3) = 4 x memo[3] = 24 memo = 2



1 - 1, 2, 3, 5, ...

Function 1 base
Array Memo 0 base

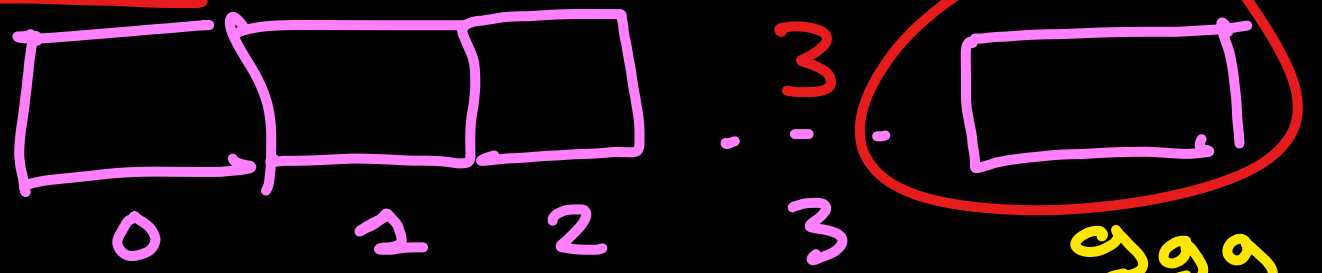
$f(1) = 1$, $f(2) = 1$, $f(3) = 2$

....

Memo i = Fibo - i

$1 \le i \le 10^3$

Memo [1000] - 1

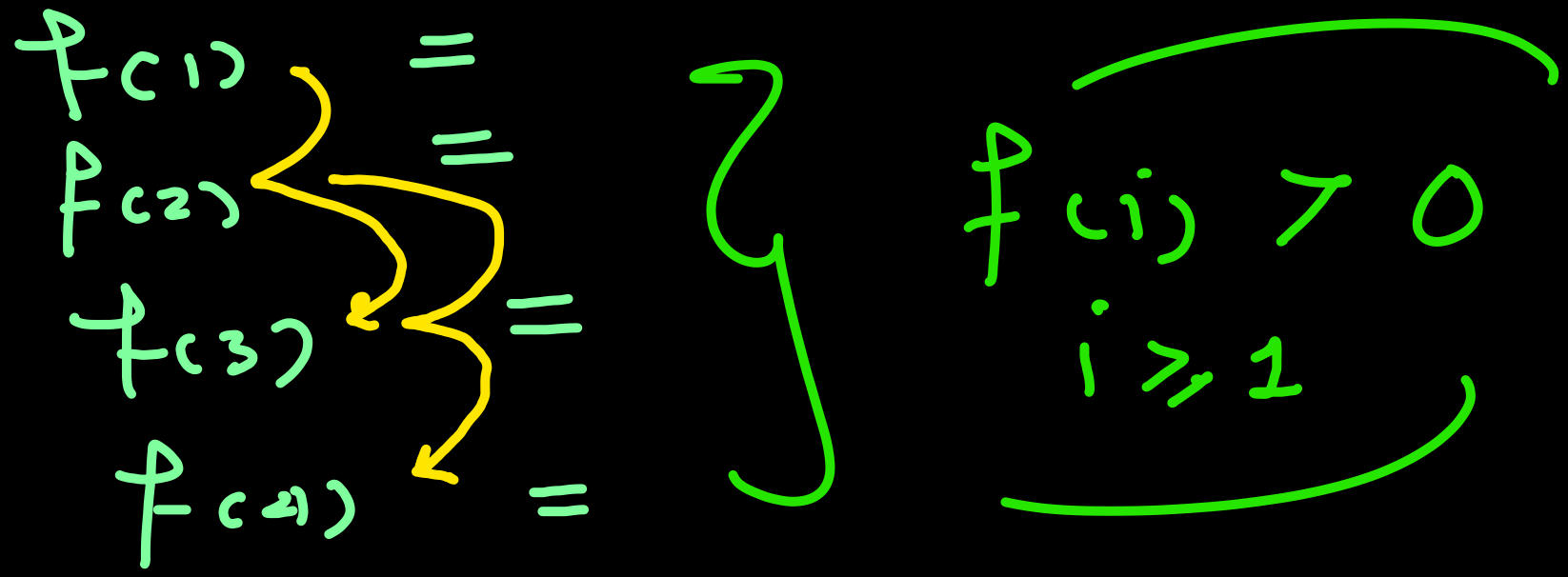


Fibo - 3 = 3

Fibo - 1000 = Memo - 1000

Memo[1005] - Memo[1002]

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



Rekursif Periodik

* Tribonacci

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

$f(2) = 0$

$$f(n) = f(n-1) + f(n-3)$$

order : 3

$$f(n) = f(n-1) + f(n-2) + \cdots + f(n-k)$$

k_1 k_2 k_i

$n - k$, $k_{max} = order$

f(3) = 0 → memo[3] = 0

f(4) = 1

f(5) = 1

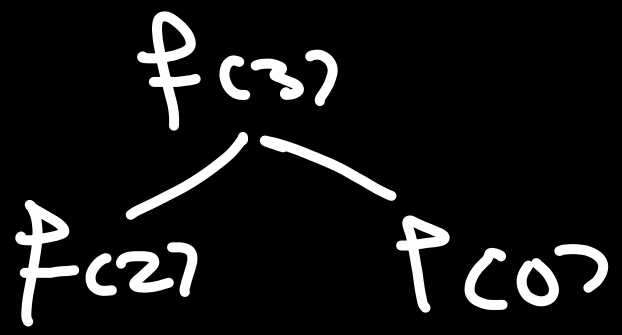
---

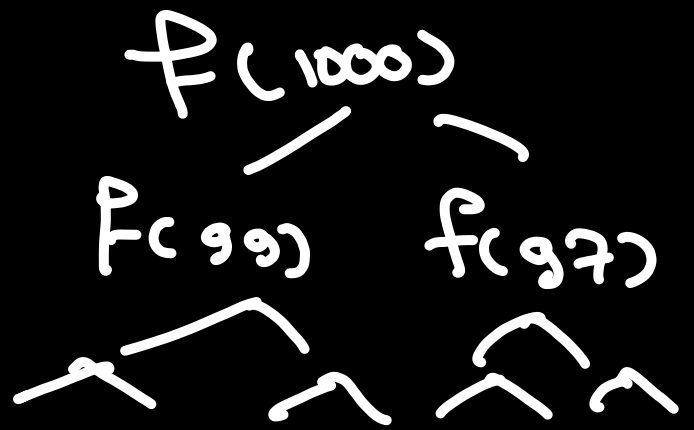
f(1000) = 0

memo[1000] = 0

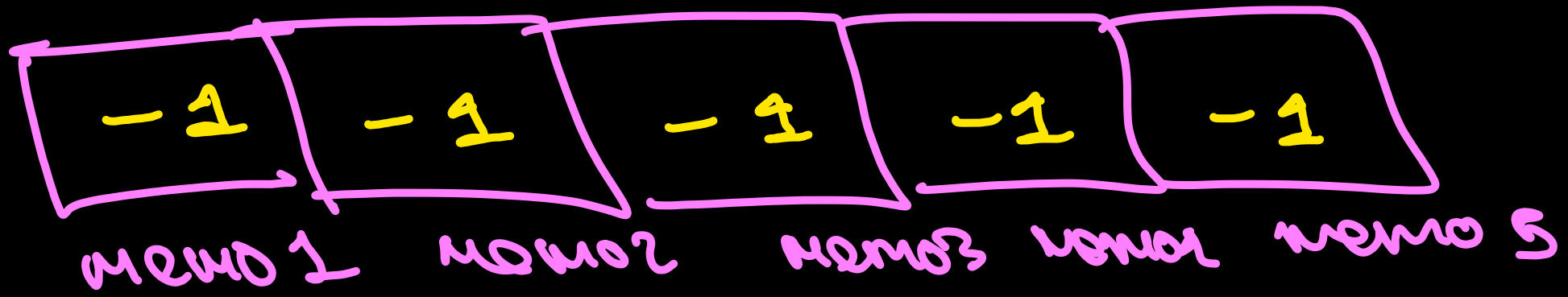
if (memo[i] != 0) sudah dihitung

else (memo[i] == 0) blm dihitung.





$F(i) = 0$



* Dynamic Programming

$f(x, y)$ = Total item max dari $(1, 1) \rightarrow (x, y)$

* Top - down
* Bottom - up

* bawah, * kanan

(r, c) —bawah→ $(r+1, c)$

(r, c) —kanan→ $(r, c+1)$

* Prefsum
* For loop

↑

* Bottom up

* Top - down (rekursif)

Bottom-up : * bawah , * kanan

Top - down : * atas , * kiri

(r - 1 , c) , (r , c - 1)

Setiap borg erak < atas, kiri } Max

Max (atas , kiri)

$$f(x, y) = item[x][y] + \max(f(x-1, y), f(x, y-1))$$

$f(x,y) = item[x][y] + \max(f(x-1,y), f(x,y-1))$

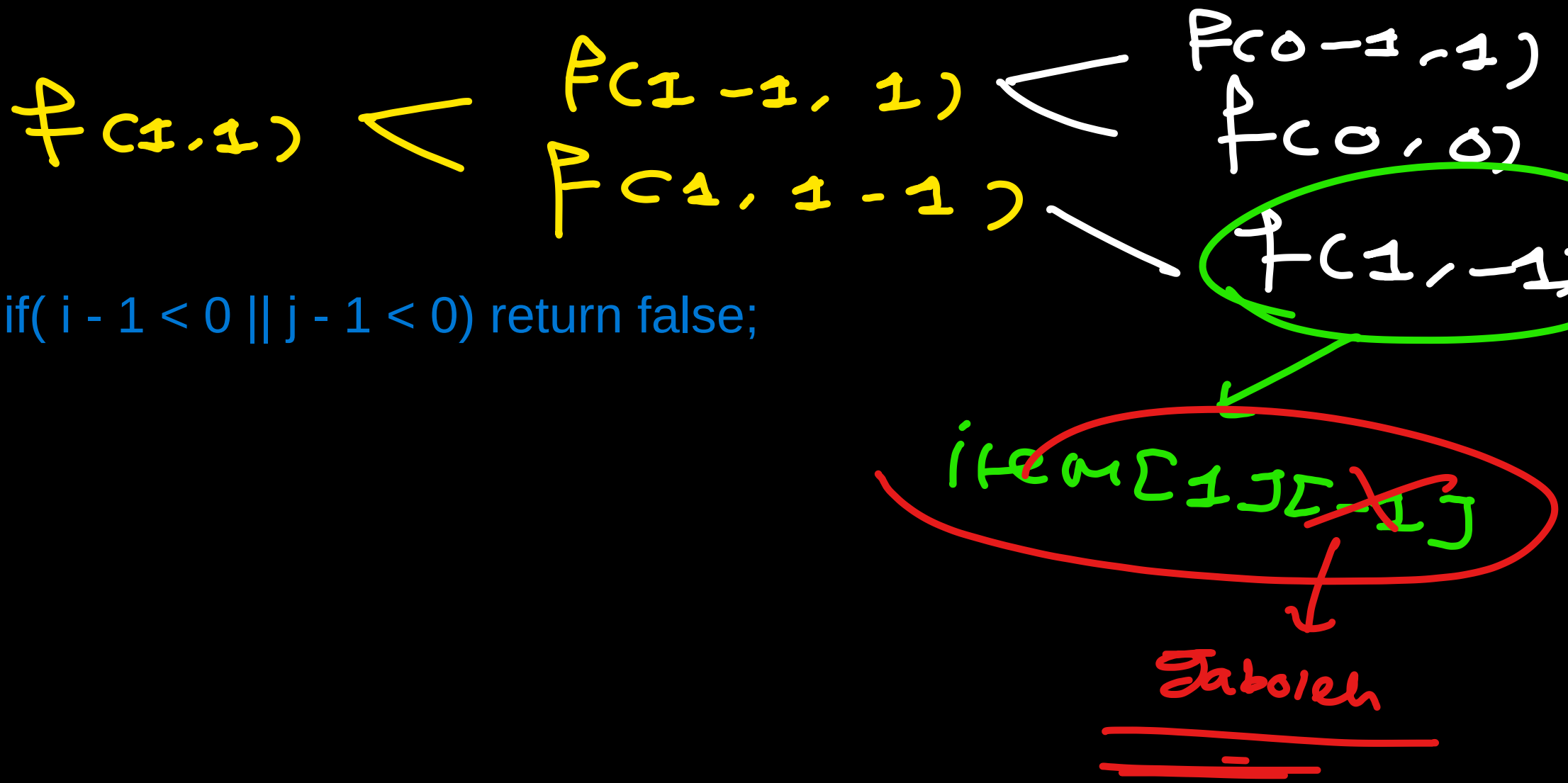
Mengambil item sebanyak

+ Base cases

$f(1,1) = item[1][1]$

$f(1,2) = item[1][1] + item[1][2]$

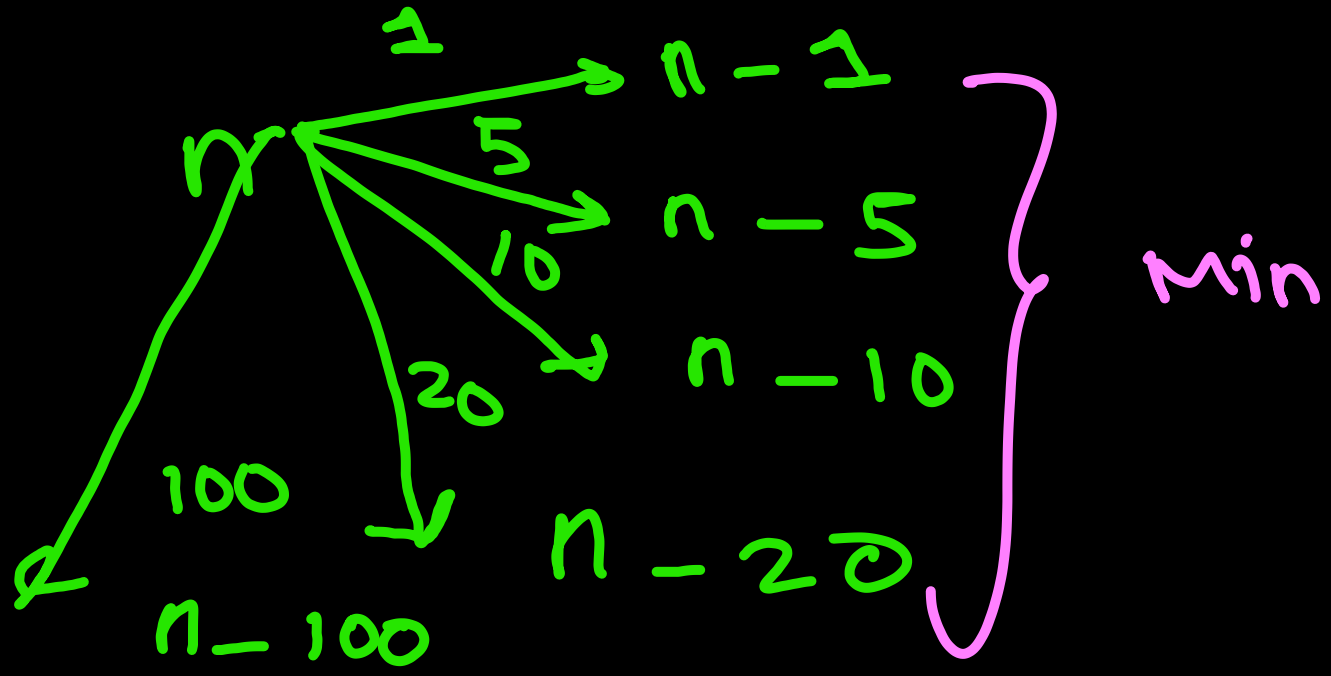
$f(2,1) = item[1][1] + item[2][1]$




```
if( i - 1 < 0 || j - 1 < 0) return false;
```

$f(n)$ = Jumlah nominal min yg dipermacan untuk uang sebesar n

1, 5, 10, 20, 100



$n = 20$

nominal

1 lembar ?

bisa 19

1 Lembar

if (n < 0) ret Fail

$f(n) = 1 + \min(f(n-1), f(n-5), f(n-10), f(n-20), f(n-100))$

$f(1) = 1, \quad f(5) = 1, \quad f(10) = 1$

$f(20) = 1, \quad f(100) = 1$

f(3)