

MODUL 9

ALGORITMA REKURSIF

A

Aliza Nurfitrian Meizahra - 1101223083

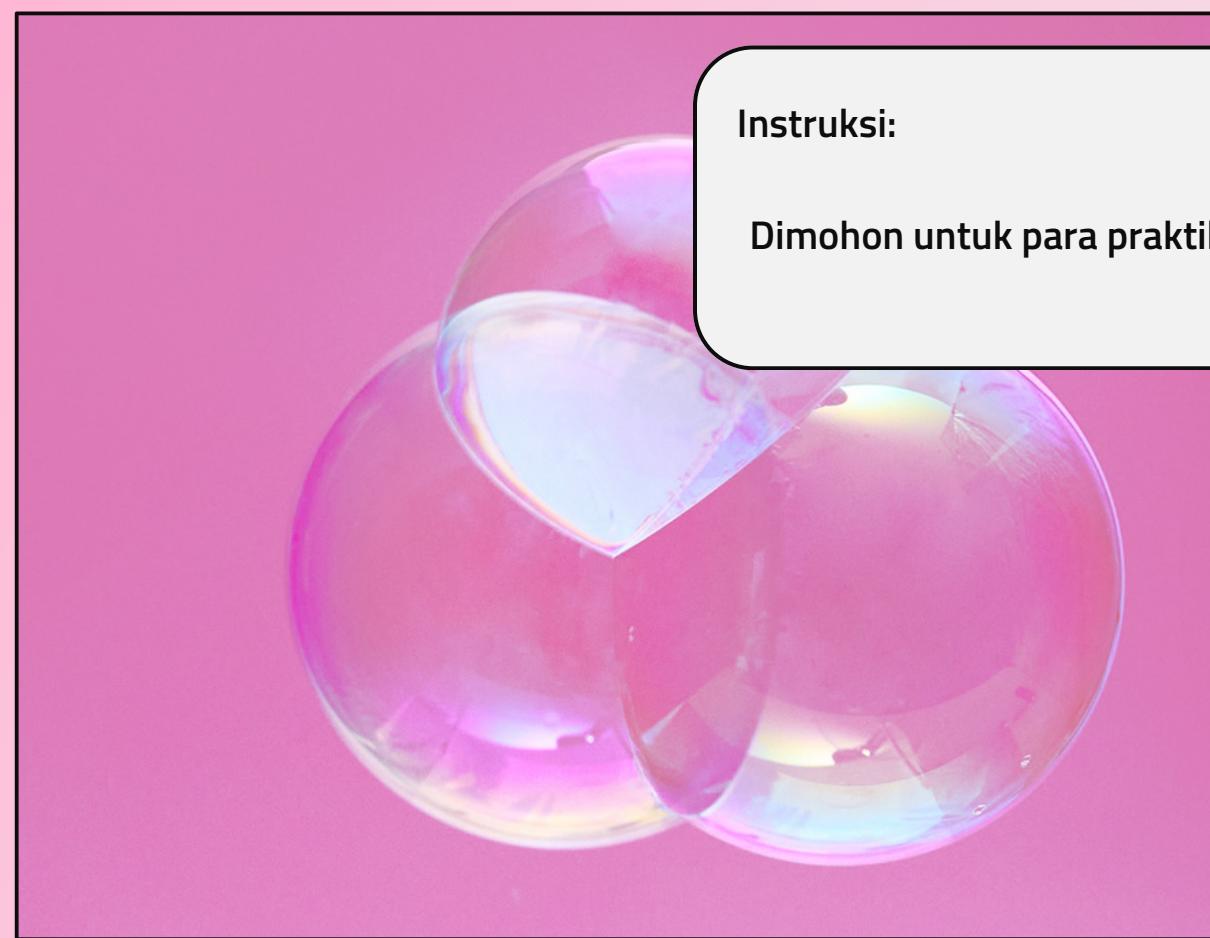
Cancel

Start

File Edit View Help

Topik Utama

img000.jpg

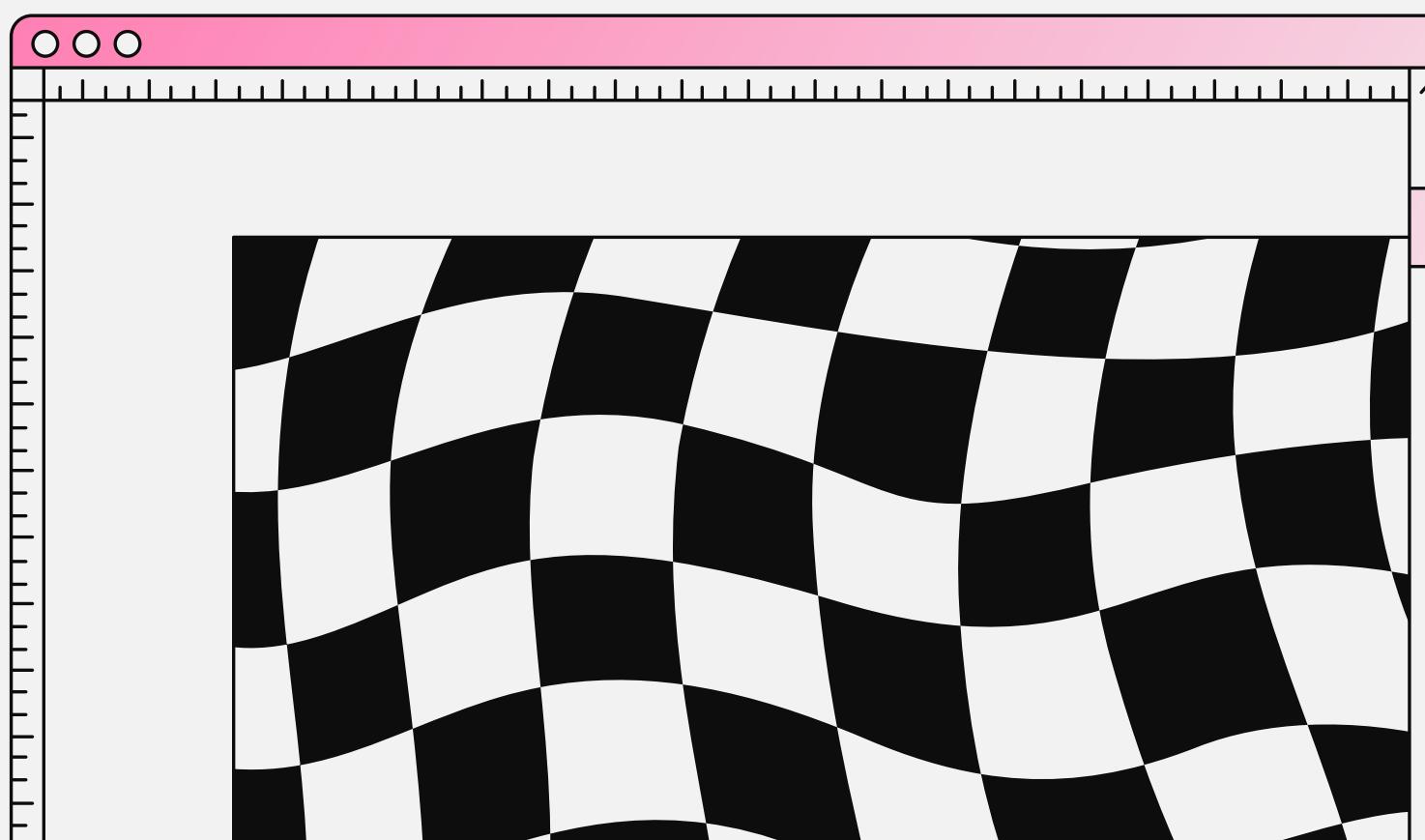


Instruksi:

Dimohon untuk para praktikan untuk memperhatikan slide power point ^_^

01	<u>Tujuan Praktikum</u>
02	<u>Definisi Algoritma Rekursif</u>
03	<u>Basis dan Rekurens ?</u>
04	<u>Kelebihan Kekurangan Rekursi</u>
05	<u>Contoh Kasus</u>

Tujuan Praktikum



Memahami prinsip kerja dari algoritma rekursif

Memahami penggunaan basis dan rekurens pada algoritma rekursif

Mengimplementasikan algoritma rekursif sebagai pengganti iteratif dalam program sederhana

The image shows a presentation slide with a light purple background. At the top center, the title "Algoritma Rekursif" is displayed in a large, bold, black font. Below the title, there is a smaller section with the heading "Rekursi" in bold. To the left of this section, a progress bar window titled "Untitled -TextEdit" is overlaid. The progress bar window has a red header bar with the word "PROGRESS" and a red "X". Below it, a white bar contains the text "LOADING..." and a red progress bar. A black "STOP" button is at the bottom right of the progress bar window. The overall interface has a purple header bar with standard window controls (minimize, maximize, close).

Algoritma Rekursif

"Rekursi"

algoritma yang melakukan pemanggilan terhadap dirinya sendiri, dimana hanya dapat digunakan pada prosedur dan fungsi

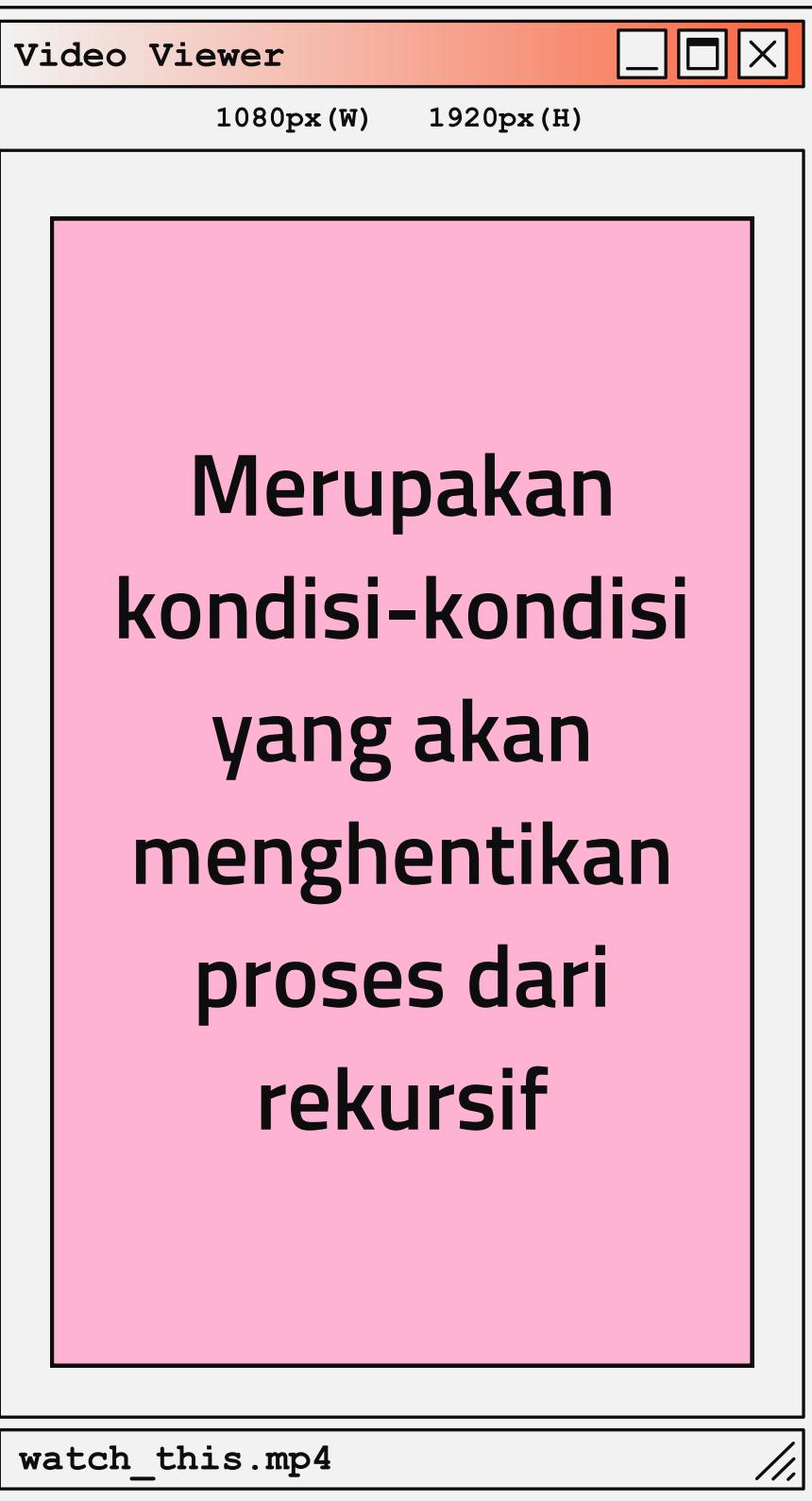


[Back to Agenda Page](#)

Basis

Untitled -TextEdit

File Edit View Help



```
#include <stdio.h>

struct mahasiswa{
char nama[20],nim[20];
} mhs;

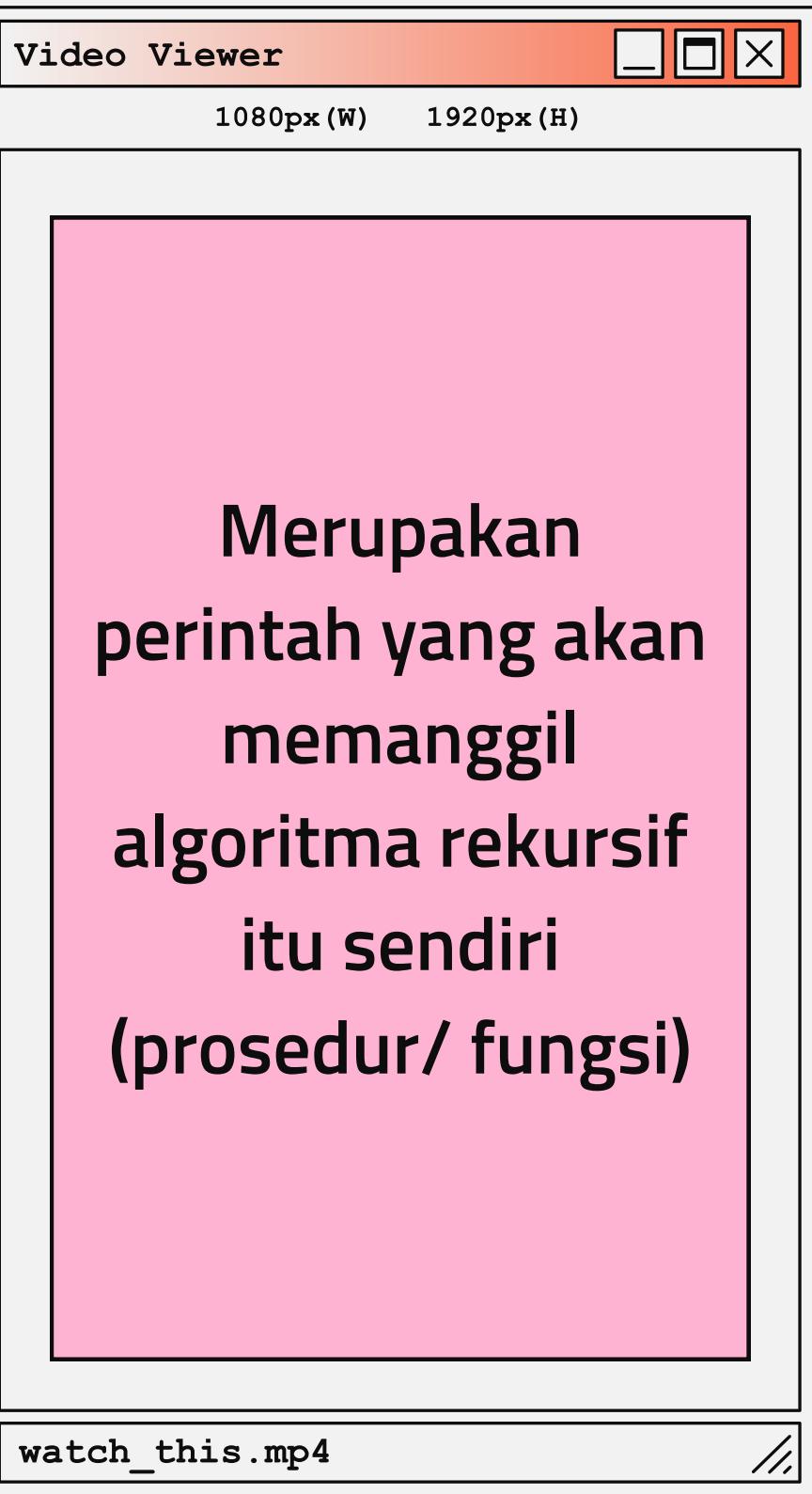
int n;
void input(int x);
int main(){
printf("Banyak Mahasiswa : "); scanf("%d", &n); getchar();
input(n);
return 0;
}

void input(int x){
if (x > 0){
printf("Nama : "); gets(mhs.nama);
printf("NIM : "); gets(mhs.nim);
x=x-1;
input(x);
}
}
```

```
int nFaktorial(int n){
if (n==1 || n==0){
return 1;
```

```
void input(int x){
if (x > 0){
```

Rekurens



Untitled - TextEdit

File Edit View Help

```
#include <stdio.h>

int nFaktorial(int n);
int x, hasil;
int main(){

    printf("FUNGSI FAKTORIAL\n");
    printf("silahkan masukkan angka: "); scanf("%d",&x); getchar();
    hasil = nFaktorial(x);
    printf("Hasil faktorial: %d", hasil);
    return 0;
}

int nFaktorial(int n){
    if (n==1 || n==0){
        return 1;
    } else if (n>1) {
        return n * nFaktorial(n-1);
    }
}
```

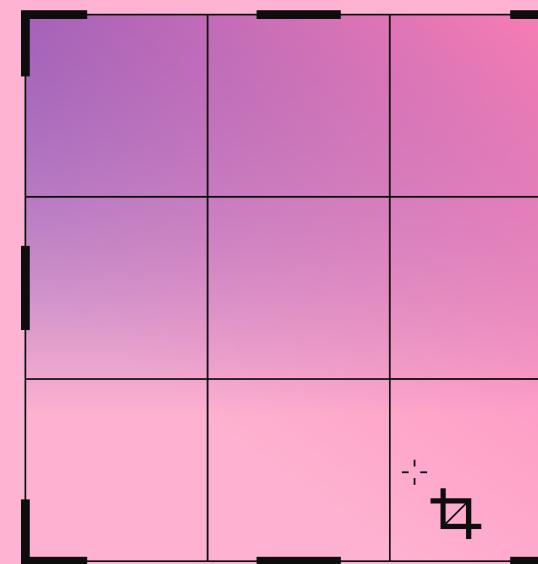
```
void input(int x){
    if (x > 0){
        printf("Nama : "); gets(mhs.nama);
        printf("NIM : "); gets(mhs.nim);
        x=x-1;
        input(x);
    }
}
```

A large pink arrow points from the recursive call in the first code block to the recursive call in the second code block.

Kelebihan Kekurangan Rekursi

Page 1

Words: 88



[Back to Agenda Page](#)

KELEBIHAN

- Dapat menyelesaikan permasalahan yang kompleks

KEKURANGAN

- Potensi Stack Overflow
- Kurang Efisien



Contoh Kasus



Program Faktorial

Program Fibonacci

Program Pangkat

New Tab

CONTOH KASUS

New Tab

CONTOH KASUS

New Tab

CONTOH KASUS



```
#include <stdio.h>

int hitungFactorial(int input) {
    if(input > 1)
        return input * hitungFactorial(input - 1);
    else
        return 1;
}

int main(void){
    int angka,hasil,i;

    printf("## Program Bahasa C Menghitung Faktorial ## \n");
    printf("===== \n\n");

    printf("Input angka: "); scanf("%d",&angka);
    printf("%d! = %d \n",angka, hitungFactorial(angka));

    return 0;
}
```



```
## Program Bahasa C Menghitung Faktorial ##
=====

Input angka: 5
5! = 120
```

[Back to Agenda Page](#)



```
#include <stdio.h>

int fibonacci(int n) {
    if (n == 0 || n == 1) {
        return n;
    } else {
        return (fibonacci(n - 1) + fibonacci(n - 2));
    }
}

int main() {
    int n, i, j = 0;
    printf("Masukkan batas jumlah bilangan fibonacci: ");
    scanf("%d", &n);
    printf("Hasil bilangan fibonacci: \n");
    for (i = 1; i <= n; i++) {
        printf("fibonacci ke-%d: %d\n", i, fibonacci(j));
        j++;
    }
    return 0;
}
```

[Back to Agenda Page](#)



```
Masukkan batas jumlah bilangan fibonacci: 5
Hasil bilangan fibonacci:
fibonacci ke-1: 0
fibonacci ke-2: 1
fibonacci ke-3: 1
fibonacci ke-4: 2
fibonacci ke-5: 3
```



```
#include<stdio.h>

int main(){
    int x,y;

    printf("Masukan bilangan yang akan dipangkatkan: "); scanf("%d", &x);
    printf("Jumlah pangkat bilangan: "); scanf("%d", &y);
    printf("Hasil %d pangkat %d = %d\n", x,y, pangkat(x,y));

    return 0;
}

int pangkat (int x, int y){
    if (x == 1 || y == 0){
        return 1;
    }
    else {
        return x* pangkat(x, y-1);
    }
}
```

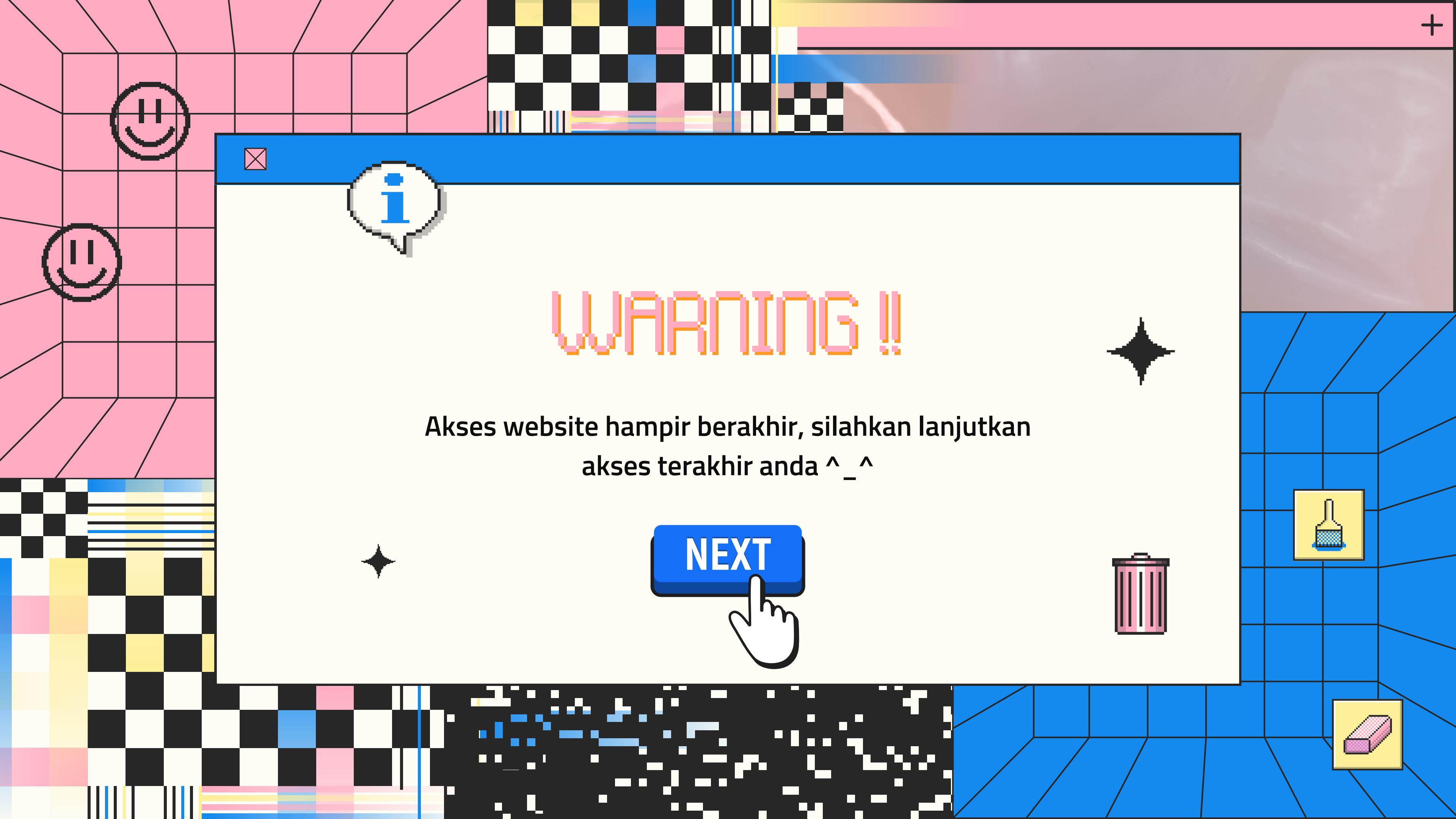
New Tab +

< > 🔎 CONTOH KASUS ⌂ ⌚

Program Pangkat

```
Masukan bilangan yang akan dipangkatkan: 5
Jumlah pangkat bilangan: 2
Hasil 5 pangkat 2 = 25
```

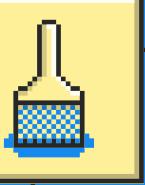
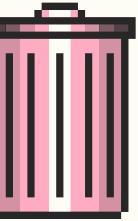
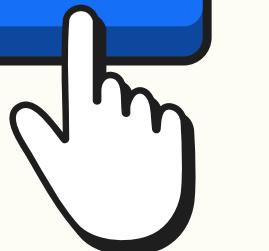
[Back to Agenda Page](#)



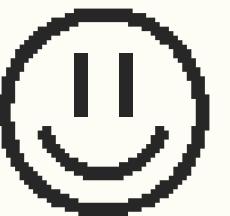
WARNING !!

Akses website hampir berakhir, silahkan lanjutkan
akses terakhir anda ^_^

NEXT



THANK YOU



Sumber:
Modul Praktikum Algoritma
Pemograman, FTE