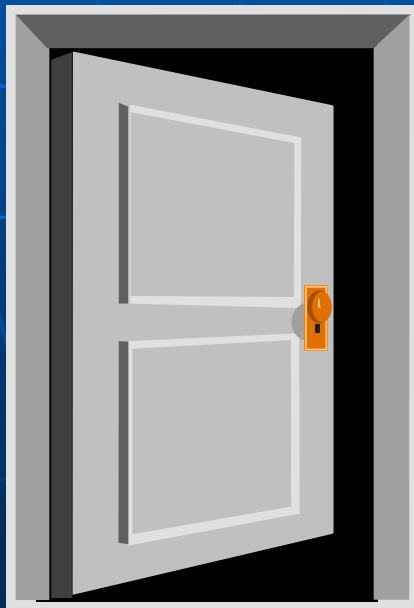
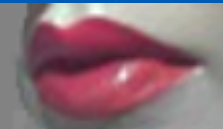


# Pengantar Algoritma Genetika

(genetic algorithm, GA)

Note-nya Cak "Tri Budi"

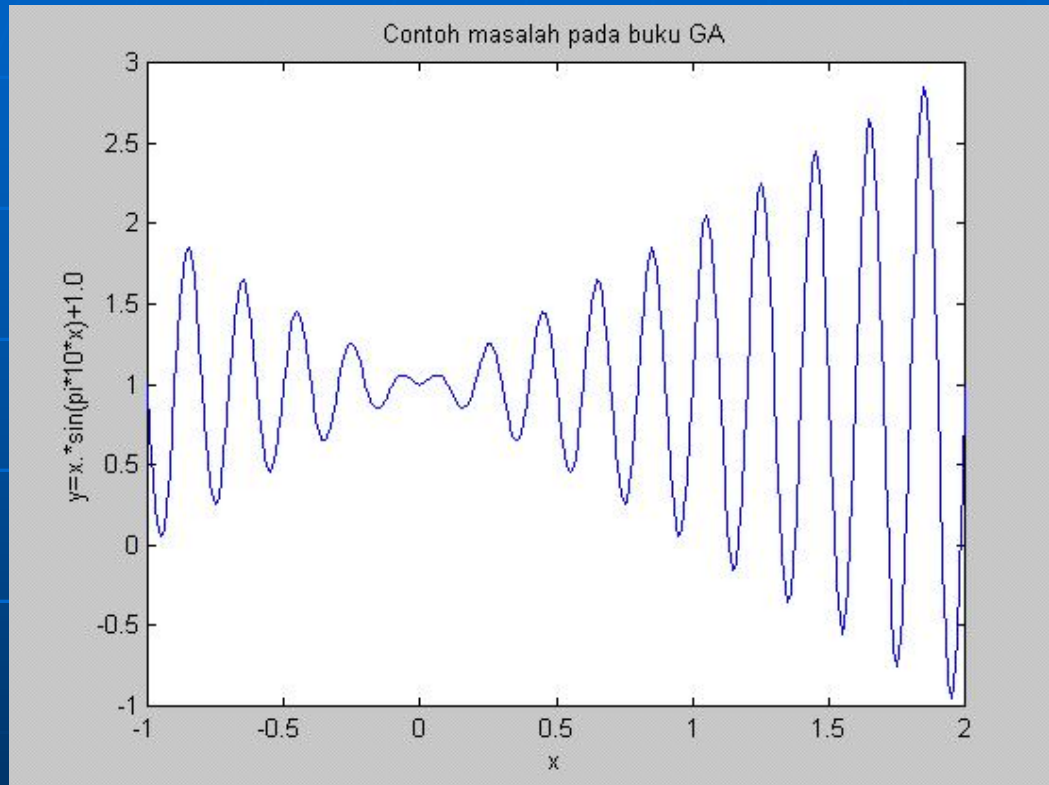
## Speech Signal Processing



Suatu metode simulasi yang didasarkan pada logika genetik.

- mencari keturunan yang memiliki fitness function terbaik
- direalisasikan dalam bentuk cromossom binary
- melalui jalan mutasi atau crossover

# Memulai dengan sebuah fungsi



$$y = x \sin(10\pi x) + 1.0$$
$$-1.0 \leq x \leq 2.0$$

Bagaimana mencari nilai  $x$  agar diperoleh nilai  $y$  yang optimum untuk keseluruhan (global optimum) ?

# Dari Sistem ke simulasi

x → diformulasikan sebagai "chromosome binary"  
y → diformulasikan sebagai "fitness function"

Range nilai x

Dalam kasus ini kita mencoba menyelesaikan dengan cara paling sederhana, yaitu menggunakan operator mutasi



Chromosome binary yang akan mewakili:

000000000000000000000000



111111111111111111111111

$$\text{Resolusi 1 bit} = (2 - (-1)) / (2^{22} - 1)$$

# Algoritma ..1

# Inisialisasi generasi ke-1 (chromosome induk)

1. Bangkitkan sebuah vektor yang mewakili sebuah chromosome 22 bit  $\rightarrow b[0]$  s/d  $b[21]$
2. Konversikan ke bentuk desimal, dalam hal ini dari base-2 menjadi base-10, dan dituliskan sebagai  $x'$
3. Cari nilai  $x$  sebenarnya:  
$$x = -1.00 + x' \cdot 3.00 / (2^{22} - 1)$$
4. Evaluasi fitness function-nya  
$$y = x \sin(10 \pi x) + 1.0$$

# Mencari generasi baru.....

# Algoritma ..2

# Mencari generasi baru yang memiliki fitness function optimum,  $f(x)_{\max}$ :

1. Bangkitkan chromosome baru dengan cara mutasi  
Pilih nilai  $i$  secara acak, dan rubah nilai  $b[i]$  dari 0 menjadi 1, atau dari 1 menjadi 0
2. Konversikan ke bentuk desimal  $\rightarrow x'$
3. Cari nilai  $x$  sebenarnya
4. Evaluasi fitness function-nya  $\rightarrow f(x)$   
Bandingkan dengan nilai  $f(x)$  sebelum mutasi  
Jika  $f(x)$  baru  $>$   $f(x)$  lama, maka  $f(x)$  baru diterima  
Jika tidak, maka  $f(x)$  baru tidak diterima dan kembalikan chromosome seperti sebelum mutasi
5. Ulangi proses mutasi ini sampai 100 kali, 1000 kali atau terserah anda sampai anda puas dengan fitness function maksimum yang didapatkan

# Program..1

disini menggunakan Turbo C 2

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>

int i,j,ii=22;
int k,kk=100;
int b[24],b_i,dec_i,dec;
int b_new;
float pi,x,x_1,y;
float gen_0,gen[10];

/*pembangkitan chromosom induk*/
void crhomosom()
{
printf("\n");
for(i=1;i<=ii;i++)
{
b[i]= (int)rand()%2;
}printf("\n");
}

/*konversi binary ke desimal*/
void bin_to_dec()
{
for(i=1;i<=ii;i++)
{
dec_i=0;b_i=b[i];
for(j=0;j<i;j++)
{
b_i=b_i*2;
}
dec_i=b_i/2;
}
}

/*fungsi yang dicari*/
void sin_func()
{
x_1=(float)dec;
x=-1.0+x_1/(419404-1);
pi=acos(-1.0);
y=x*sin(10*pi*x);
}

```

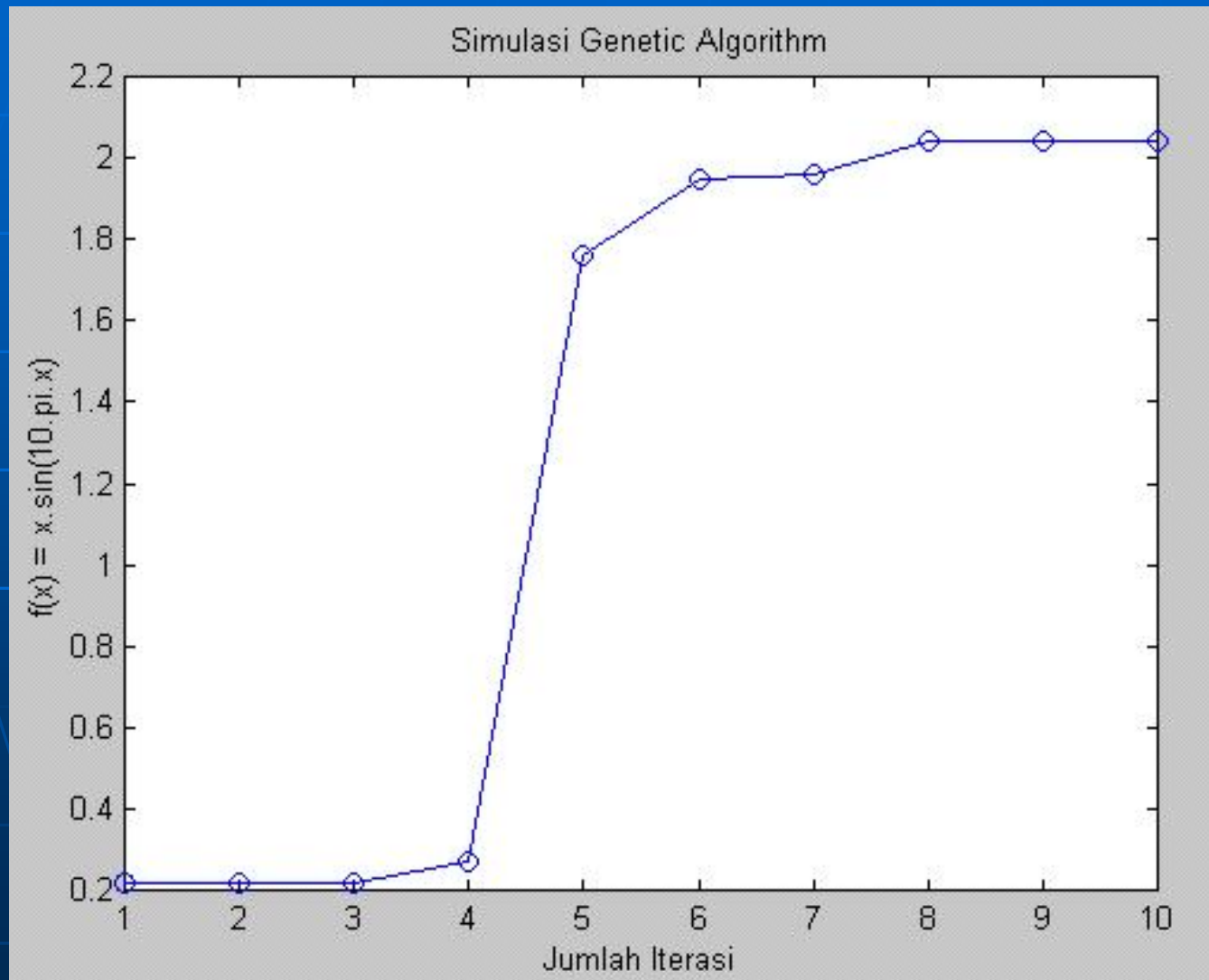
# Program..2

```
/*mencari generasi terbaik melalui mutasi gen*/
void mutasi()
{
    gen_0=y;
    printf("\ngenerasi induk  =%6.5f",y);
    for(k=1;k<=kk;k++)
    {
        i=(int)rand()%23;
        b_new=(b[i]+1)%2;
        b[i]=b_new;

        bin_to_dec();
        sin_func();
        gen[k]=y;
        /*penentuan generasi baru diterima atau tidak*/
        if(gen[k]>gen_0)
            {gen_0=gen[k];}
        else
            {
                gen_0=gen_0;
                b[i]=(b_new+1)%2;
            }
        printf("\nx=%f fitness gen[%d]:%f",x,k,gen_0);
    }
}

/*Program utama*/
void main()
{
    crhomosom();
    bin_to_dec();
    sin_func();
    mutasi();
    return 0;
}
```

# Hasil-hasil Sampai dengan 10 iterasi





# Hasil-hasil Sampai dengan 100 iterasi

