Kelompok 3

* Dewi Nurisma (H1A024002)
* Dwima Agung P (H1A024020)
* Bagus Satria Bima Sakti (H1A022

1. Penjelasan bagaimana Discrete\_convolution\_plot\_step.m bekerja ?

Jawaban :

1. Sinyal awal

x = [1 2 3] → sinyal input.

h = [1 1] → respon impuls.

Panjang hasil konvolusi = N = length(x) + length(h) - 1 = 4.

2. Looping per langkah n

Program jalan dari n = 0 sampai n = N-1.

Di tiap langkah, h dibalik dulu (fliplr) → ini bagian h[-k].

Lalu h digeser sesuai nilai n → jadi h[n-k]

3. Cari overlap & kalikan

Bagian x dan h yang ketemu (overlap) dikalikan satu-satu.

Hasil perkalian ini ditampilin di subplot ketiga.

Semua hasil kali dijumlahin → jadi nilai y[n].

4. Visualisasi step by step

Subplot 1: sinyal x[n] asli.

Subplot 2: h yang dibalik & digeser.

Subplot 3: hasil kali per elemen (x \* h).

Judul subplot nunjukin jumlah perkalian yang jadi nilai y[n].

5. Animasi

Ada pause(1) → jadi tiap langkah konvolusi kelihatan kayak animasi.

Setelah loop selesai, kita punya y = [1 3 5 3].

1. Penjelasan bagaimana Discrete\_convolution\_plot\_step\_2.m bekerja ?

Jawaban :

1. Bikin sinyal dulu

x[k] = pulsa kotak dari k=0 sampai 4.

h[k] = sinyal decaying (0.8)^k dari k=0 sampai 6.

2. Setup konvolusi

Tentuin rentang index hasil (n\_start sampai n\_end).

Balikin h jadi h[-k], biar bisa digeser sesuai n.

3. Looping tiap n

Untuk setiap n, geserin h[-k] → jadi h[n-k].

Cari bagian x[k] yang overlap sama h[n-k].

Kalikan yang ketemu, lalu jumlahkan → dapet y[n].

4. Tampilan step-by-step

Ada 4 subplot:

1. x[k] asli

2. h[n-k] yang digeser

3. Hasil perkalian titik-titik (product)

4. Hasil akumulasi y[n] sejauh ini

Judul (sgtitle) nunjukin rumus perhitungan: apa aja yang dijumlahkan untuk dapet nilai y[n].

5. Akhirnya

Ditampilkan sinyal y[n] lengkap.

Dibandingin sama hasil conv(x,h) bawaan MATLAB → hasilnya sama.

1. Penjelasan bagaimana Discrete\_convolution\_plot\_step\_2.m bekerja ?

Jawaban :

1. Definisi sinyal

x = [1,2,3,1,0,0] dan h = [1,1,2,0,0,0] adalah sinyal diskrit.

Indeks keluaran n dihitung dari jumlah indeks awal hingga jumlah indeks akhir → panjang hasil = length(x)+length(h)-1.

2. Persiapan pergeseran

h dibalik dengan fliplr(h) untuk mendapatkan h[-k]. Dengan menambahkan current\_n, didapat bentuk geser h[n-k].

3. Perhitungan tiap n

Untuk setiap nilai n, dicari bagian x[k] yang overlap dengan h[n-k].

Elemen-elemen yang overlap dikalikan (x[k]\*h[n-k]).

Hasil perkalian dijumlahkan → itulah y[n].

Proses ini ditampilkan dalam bentuk tabel (menunjukkan nilai x[k], h[n-k], produk, dan hasil penjumlahan).

4. Plot hasil akhir

Setelah semua n dihitung, hasil konvolusi y[n] ditampilkan dengan grafik stem.

Untuk validasi, ditambahkan juga hasil konvolusi dengan fungsi bawaan MATLAB conv(x,h). Jika benar, kedua grafik akan sama.