Pretest VIII	No. Date. / /
Auria Nanda	129:1923
2100018296 (F)	(min) scale
	m 03 () 101
66 n 93	1 -> (Dt .
1. Algoritma menampung data pada	array 2 dimensi,
melakukan perhitungan data dan me	enamprikan data
pada tampilan 2 dimensi.	ref Las
/ Menainputkan jumlah baris k ko	olom pada matriks
dengan tipe data integer, law men	nggunakan perulangan
untuk menampilkan nilai matriks	(untuk kedua matrits)
dan menampilkan nilai matriks ke	daiam bentuk tampiran
2 dimensi serta melakukan penjul	miahan pada kedua
matrice denaan permangan bersan	rang dan satu
penempung ynitu hasii dengem	tipe data integer je
Y LEGICIG 2 - Zerison)	h have been a second
Deklarasi :	19.20
iij, m, n: integer	10) 600
matriks 1 : array [20] [20] of i	M SS 1 101
matries 2 : array [20] [20] of	in to 30t
hasin: array [20] [20] of int	

No. Date. Deskripsi : Read (min) for i < 1 to m do for j < 1 to n do Read (motrits 1 [i][i]) end for memory and could represent the second end for for i < 1 to m to for < 1 to n do write (matrix [i][j]) end for much as a second as a second and a second a second and a second a second and a second and a second and a second and a second an and for medianes as a second second second second for i ←1 to m do for j < 1 to n do Read (matrices 2 [i] [j]) end for end for for i ← 1 to m do for J ←1 to n do write (matrixs 2 [i][j])

for i < 1 to m do

for j < 1 to n do

hasil [i][j] = matrixs [i][j] + matrixs 2 [i][j]

write (hasil (i][j])

end for

and for.

```
Date.
                 Minaude Liostream>
                 using namespace std;
                    int main () Land 1910 and 1910
                                              int matrixs 1 [3][3] = ( <1.2.5), <39.7>, <5,6.0} :
                                            int matrixs 2 [3][3] = { \(\ia_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma_1\sigma
                                              int i, hast, j, total = 0;
                                           cout << " Menampiran matrixs A";
                                             for (i=0; i23; jtt) {
                                                                                    for (j=0; j 43; j+1) (
                                                                                                                       cout ZZ matriks1 [i][j] ZZ " ";
                                                                                        7
                                                      1
                                                          for ( i =0; i23; i++) {
                                                                                                  for (j=0; j 13; j+1) d
                                                                                                                               cout 12 matrits 2 [i] [j] 21 " ";
                                                                                                     7
                                                                                            cout & endi ;
                                                           3
                                                            cout LL "hasin penjumnahan matrits: 'n";
```

9

0

REMEMBER · MEMORY

	No.	
	for(i = 0; i23; i++) 1	
0	for (j=0; j 23; j++) x	
	hasil [i][j] = matrixs1[i][j] + matrixs2[i][j];	
	cout << hosin [i][j] << "\t";	
)	
	cout LC endi;	
	Y	
	cout LL total;	
0		
0		
0		
0		
0		
	Weng	