Ada 4 cara melakukan sortir Array dalam php.

1. Menggunakan Max.

Langkah-langkah pembuatan.

* Buatlah sebuah function yang akan digunakan untuk menyimpan perintah pembuatan tersebut.

Misal :  function counting ()

* Kemudian parameter kosong pada fun tersebut akan diisi oleh sebuah variabel array.

Ex :  function counting($array)

* Kenapa diisi variabel array? Karena kita akan menggunakan var tersebut untuk melakukan operasi di dalam fun.
* Lalu buatlah sebuah varibel baru yang akan di gunakan untuk menyimpan func max, untuk mencari nilai terbesar dari variabel array.

Ex :  $max=max($array);

Spill dikit nilai variabel array adalah sebagai berikut :  $array=[4,2,2,8,3,3,1];

* Nah, jadi, kita mendapatkan angka delapan sebagai nilai terbesar yang dimiliki array, kenapa 8 ? karena max mencari nilai terbesarnya.
* Setelah itu, buatlah variabel baru, yang akan digunakan untuk membuat Array baru.

Ex : $count=array\_fill(0,$max +1,0);

Variabel count-lah, yang terpilih untuk mengemban tugas tersebut. Lalu apa itu array-fiil? Array-fiil adalah sebuah func yang digunakan untuk membuat array baru. Di dalam array-fiil, kita membutuhkan tiga parameter yang harus diisi.

Parameter pertama berisi : indeks untuk memulai array baru, nah, kitamenggunakan angka 0 sebagai indexsnya, yang artinya kita akan memulai indeks dari 0 untuk array count ini.

Parameter kedua berisi : panjang array yang diinginkan, kita menggunakan $max+1, yang artinya, kita menggunakan panjang dari max, berapa? Yaitu 8 kemudian 8 ini ditambahkan 1. Jadi, 9.

Parameter ketiga berisi : nilai yang akan dimasukan pada array count tersebut, jadi, array count itu nilainya, akan berisi angka 0. Kira kira seperti ini $count=[0, 0 , 0, 0,0,0,0,0,0].

* Setelah itu kita akan melakukan iterasi pada array count, mrnggunakan foreach.
* Foreach akan diisi oleh value/nilai pada variabel array.

Ex :  foreach ($array as $value) {

            $count[$value]++;

        }

Mari jelaskan!!

Foreach tersebut berisi dua parameter, parameter pertama itu variabel array dan variabeel kedua adalah variabel value, yang akan mempresentasikan nilai dari variabel array. Kemudian, kita akan menjalankan variabel count dengan $value didalamnya, dan setiap iterasi akan di tambahkan.

*Maksudnya bagaimana? Teing, aku ge lier🥲. Mari jelaskan saja, jika penjelasanku benar maka aku sudah ngerti, jika salah wasalam, lihat saja tutorial/dokumentasi lainnya🙏. Terus bagaimana cara menilai penjelasanmu benar/ tidaknya? Salin dokumentasi ini, terus tanya chatgpt/ai lainnya. Kalau tidak tanya yang lebih tahu saja.*

Jadi, value pada variabbel array akan digunakan sebagai indeks dalam $count

$array kan [4,2,2,8,3,3,1] ini akan digunakan sebagai indeks dari 👇

$count kan [0,0,0,0,0,0,0,0,0]

Mengapa$count[$value]++. Berarti kita menambahkan 1 ke dalam nilai array count pada indeks yang sesuai dengan value tersebut.

* Misal, ketika $value itu 4, kita melakuan $count[4]++. Artinya kita ingin menambahkan 1 pada indexs ke-4 di array count.

Lah, kenapa di tambahin 1 bukannya 4?

Karena pada awalnya, $count dimulai dengan 0, dan saat kita temukan angka 4, kita menambhakannya 1 kali di $count tepat pada indeks ke 4. Jadi, frekuensi kemunculan angka 4 adalah 1 yang artinya 4 muncul sekali di array asli.

Intinya :yang dilakukan $count[$value]++ adalah menghitung berapa kali suatu angka di $value muncul dalam $array Ketika angka pertama kali muncul, maka pada posisi indeks yang sesuai, kita menambahkannya menjadi 1, yang mengindikasikan bahwa angka tersebut muncul sekali.

* Kita sudah memiliki $count=[0,0,0,0,0,0,0,0,0,] maka setelah $count[4]++, array count menjadi [0,0,0,0,1,0,0,0,0].
* Kenapa angka 1 muncul di indeks ke 5 ? karena indeks count di mulai dari angka 0 jadi wajar dong angka 4 di taruh di indexs ke 5.
* Kemudian value yang ke 2 = 2 jadi di indeks count yang ke 2 maka diisi 1.
* Kemudian value yang ke 3 itu = 2 maka di indeks count nya kita tambahin satu lagi di indexs ke 2 jadi yang tadinya [0,1,0,0,0,1,0,0,0,0] itu menjadi seperti [0,1,2,0,1,0,0,0,0,]. Lah kenapa ga tambahin 1 di indeks ke 3? Kan itu valuenya ada di indeks ke 3? Itu kan sudah di bilang angka yang ada di value sesuaikan dengan indeks yang ada di countnya kalau misalnya angka 2 ada di indeks ke 3 dalam value maka angka 2 ini akan ada di indeks ke 2 dalam count. Kalau misal si countnya sudah mempunya nilai, kaya tadi 1 maka tambahin lagi 1 kan 2 nya ada dua kali. Begitulah seterusnya. Semoga mudah di pahami. Dan hasil akhirnya akan menjadi seperti ini [0,1,2,2,1,0,0,0,1]
* Setelah itu kita akan membuat sebuah var baru yang berisi array kosong yang akan digunakan untuk menghasilkan array yang ber urut.
* Misal kita menggunakan $urutkan untuk membuat array kosongnya, kemudian melakukan loop.

Ex :  $urutkan=[];

Lalu lakukan loop :  foreach ($count as $value => $frekuensi) {

            for ($i=0; $i < $frekuensi; $i++) {

                $urutkan[]=$value;

            }

        }

Dalam kode tersebut, terdapat dua loop yaitu for dan foreach.

1. Foreach bertugas untuk mengiterasi array count => array yang menyimpan frekunesi itu.

$value adalah indexs dari array count. Jadi, value = [0,1,2,3,4,5,6,7,8]

$frekuensi adalah nilai dari array count sendiri. Frekuensi=[0,1,2,2,1,0,0,0,1]

1. For yang ada dalam foreach digunakan untuk mengulang nilai berdasarkan frekuensi yang ada di array count. Nilai apa yang di ulang? Yaitu nilai $value. Jadi, kalau misal nilai valuenya 2 dan di indekss ke 2 pada frekuensi itu 2 maka for akan mengulang angka 2 pada value itu 2 kali sesuai ada yang di frekuensi, namun jika nilai dalam value itu 5 dan di indexs ke 5 dalam frekuensi itu 0 maka for tidak akan mengulang apapun.

Untuk lebih jelasnya :

**Proses**

Value =0, frekuensi di indeks ke 0 = 0. Tidak di ulang karena nila = 0

Value =1, frekuensi di indeks 1 = 1. Di ulang satu kali.

Value = 2, dan frekuensi di indeks 2 =2 maka di ulang 2 kali karena frekuensi nya ada 2

Value =3 dan frekuensi di indeks 3 = 2 maka di ulang si 3 nya itu dua kali

Value = 4 dan frekuensi di indeks 4 = 1. Maka di ulang 1 kali

Value = 5 dan frekuensi di indeks 5=0 maka tidak ada perulangan

Value = 6 dan frekuensi di indeks 6=0 maka tidak ada perulangan.

Value =7 dan frekuensi di indeks 7 = 0 maka tidak ada perulangan

Value =8 dan frekuensi di indeks 8 =1 maka di ulang satu kali

Lalu nilai nilai itu akan dimasukan ke dalam variabel urutkan. Jadi, $urutkan =[1,2,2,3,3,4,8]

Tambahan : jadi pakai <frekuensi itu gini ya apakah i kurang dari frekuensi? ya, mak jalankan. kalau misalnya apakah i = frekuensi tidak maka jangan jalankan. Begitu, lalu kenapa yang frekuensinya 2 itu jalannya 2 kali dan bukan sekali, sementara kan <frekuensi/ <2 saat frekuensinya = 2. Itu karena, si $i itu mulainya darri 0 jadi 0,1 itu dua kali iterasi.

 return $urutkan;

jadi di return itu artinya sudah di jalankan namun msih di simpan belum di tampilkan.

Kemudain tampilkannya disini

$array=[4,2,2,8,3,3,1];

    echo"array asli 👉 : ";

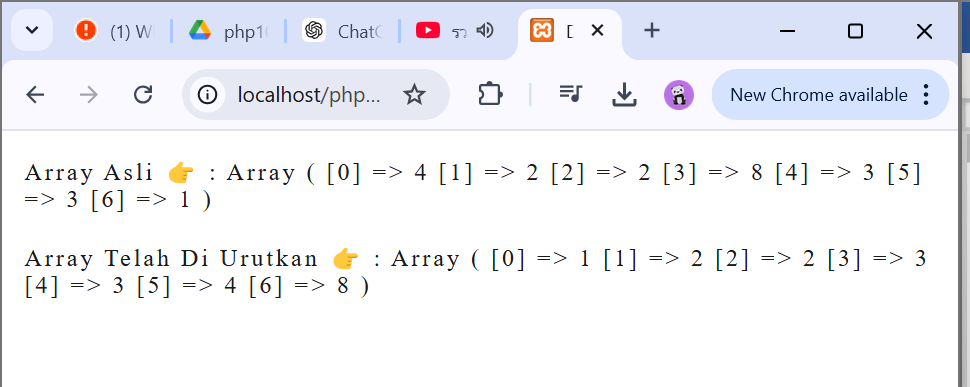
    print\_r($array);

    $urutkan =counting($array);

    echo '<br><br>array telah di urutkan 👉 : ';

    print\_r($urutkan);

hasilnya :



**Maxsimum sort methode**

Seperti biasa, mari buat func dan 1parameter(yang ingin di urutkana) dulu untuk menyimpan perintahnya :

function maxsort($Array){

* Lalu buat sebuah variabel bebas namanya mau apa saja, yang akan di gunakan untuk menghitung panjang array. Nah, untuk menghitung panjang array sendiri bisa pakai func count ya lalu parameternya isi dengan variabel array yang mau di hitung.

Ex :  $length =count($Array); // 7

Jadi operasi count tersebut di simpan dalam variabel length ya.jadi karena nilai pada var Array itu ada 7 jadi si count otomatis 7.

* Buat sebuah loop ya, tujuannya untuk mengurutkan, untuk penjelasannya saya akan menjelaskan dengan analogi saya sendiri.

Ex :  for ($i=0; $i < $length - 1; $i++) {

       $maxindex=$i;

       for ($j=$i + 1; $j < $length ; $j++) {

jadi pada intinya kode di atas itu, menjelaskan seperti ini. ini menurut saya ya, kalau beda analogi ya berarti beda jangan maksain untuk sama.

Pertama kodenya itu kalau di baca seperti ini, i = 0 i kurang dari length-1/6 maka i ++ kemudian buat var baru untuk menyimpan si i ini lalu jalankan j=i+1 dan j kurang dari length lalu j ++

Artinya begini, i semisal adalah indeks untuk si array asli ya, jika i = 0 berarti dalam array pada indekss ke 0 adalah 64 misal, jadi i = 0 sama saja kaya i = 64, nah karena kita mau mengurutkan kan, jadi si i ini hanya nilai pembanding saja dan belum ada nilai yang membandingkannya, maka j lah yang akan menjadi nilai pembandingnya. Seperti pada kode di atas j=i+1 jadi kalau i=0 maka j = 1 jika i = 64 maka j =34. Nah si i kan masih lebih besar dari j maka perulangan akan false dan akan terus berlanjut sampe menemukan bahwa si i < j, nah, kalau sudah i < j nilai true. Loop di hentikan dan mereka melakukan swap.

Begini ceritanya :

I yang saat ini cosplay menjadi 64, loop yang sedang ia lakukan tiba-tiba terhenti tepat pada indexs ke 6, ia tak bisa lagi terus melangkah maju karena j yang saat ini cospaly menjadi 90 lebih besar darinya, yang membuat nilai keduanya true.

I mematung di tempat tepat pada indeks ke 5, sembari menatap lurus ke arah j yang sedang berbaring sambil membaca sebuah buku. menyadari sepasang mata sedang mengawasinya, j segera berbalik menatap balik ke arah i.

I tersenyum ke arah j.” hallo,” sapa i padanya, senyum tersungging di bibirnya.

“Apa?” tanya J, bibirnya menguap karena kantuk yang melanda. Kemudian mengarahkan kembali pandangannya pada buku.

“Tuker tempat yuk, j!” ajak i pada j. dengan harapan j mau menerima tawarannya tanpa embel-embel apapun.

“Kenapa?” tanya j acuh.

“kenapa?” tanya j acuh, ia membuka halaman selanjutnya dari buku itu.

Helaan napas panjang yang terdengar begitu berat. Ia berkata, “ ga tahu j gw di suruh tuker tempat sama elu. Lagian gw bosen di jadiin yang ke 0mulu, kali kali gw di jadiin yang ke 6.” Mata i berbinar. “seru kya nya , ada di posisi elu,” lanjut I.

J terdiam, seolah suara I hanya angin lewat semata.

“J, ayolah, kasihan lho, “ I memohon pada J.

“Ke elu?” tanya J

“Bukan gw yang harus di kasihani, tapi orang yang membuat cerita kini ini, dia kaga paham- paham sama konsep proses kita, akhirnya dia inisiatif buat bikin skenario sendiri, terciptalah cerita tentang gw sama elu,” terang I

“terus kenapa lu ma bantu dia?” J menggeraan tubuhnya hingga ia dalam posisi duduk saat ini.

I menggelengkan kepalanya kuat-kuat.“Siapa bilang gw mau bantuin dia, itu orng buat gw supaya mau ngertiin dia mau bantu dia mahami konsep kita.”

“ga bisa ngertiin konsep orang tapi pengen di ngertiin, egois banget,” jawab J. i mengangguk setuju atas jawaban dari J

J kembali menatap I, matanya sedikit menyipit. “ lagian lu juga Bodoh,” ucapnya dengan nada sarkastik.

“Ha?” i tak terima dirinya di panggil bodoh oleh J.

“gw sebagai indeks pembanding dan elu sebagai indeks proses, melakukan pertukaran saat loop udah selesai dan maxindex udah nemu nilainya. Semisal tadinya maxindex 0, di indeks ke 0 pada array asli adalah 64, jadi saat 64 < 90 loop true dan berhenti, maxindex ketemu nilainya yaitu indeks yang di tempatin sama gw yang saat ini cosplay jadi 90, sudah itu lu i sama gw ngelakuin swap, terus lanjut ke iterasi ke 2,” J Perlahan, ia mulai mempersiapkan tubuhnya untuk berpindah posisi. Dengan satu tarikan napas dan sedikit dorongan dari tangan kiri, tubuhnya mulai terangkat perlahan. Kaki-kaki yang tadinya terlipat mulai lurus, dan tubuhnya bergerak ke atas. Kepala terangkat, diikuti dengan punggung yang tegak, dan akhirnya ia berdiri tegap.

“Oh... kok gw baru tahu ya,jadi percuma dong ajakan gw sebelumnya,” mulut I menganga antara takjub dengan penjelasan J atau karena dirinya baru tahu konsep tersebut.

“Kan lu bodoh, mangkannya baru tahu.” J berjalan mendekat ke arah I

“Bukan gw lah yang bodoh.”

“terus? Lu mau nyalahin siapa?” J kini berada tepat di depan I

“orang yang nyeting gw sampe gw ga paham sama konsep gw sendi—“sebelum I selesai melanjutkan ucapannya, J mendorong I Masuk kedalam ruangan yang di tempati J sebelumnya yaitu indeks ke 6 kemudian J segera pergi ke indexs 0 tempat I sebelumnya berada.

Tamat

  if ($Array[$j] > $Array[$maxindex]) {

            $maxindex =$j;

        }

Apa Anda tahu apa yang sedang terjadi?

Jadi saat ini if ini akan membandingkan jika si array[j] >array[maxindex] maka maxindex=j

Jadi kan seperti tadi si maxindex itu indeks proses dan si j itu indeks pembanding. Dan seperti biasa indeks yang di maksud ini indx pada array asli kan

Jadi :

Array[1] kan tadi j itu i+1 jadi jika i = 0 maka j =1

Nah disini max indeks = i , i dimulai dari 0 maka maxindex saat ini 0

Jadi jika array[1]>array[0] ini false karena nilai indeks 1 = 34 sementara indeks 0=64 jadi false, terus ngebandingin lagi si 0 ini ke j selanjutnya array[2]= 25>array[0] false

Array[3]=12>array[maxindex]false

Array[4]=22 >array[maxinde]dalse

Array [5[=11 juga false

Array [6]=90 true jadi maxindex = 6 kenapa 6 kan dia true saat di indeks ke 6

Yang perbandingan ke dua pun begitu.

“yo bung, kalau misal if sama sama membandingkan kaya si loop, ngapaojn di tulis. Kan semua proses dari perbandingan dari yang terkecil ke terbesar dn swap tempat sudah di lakuin di loop, apa lu tulis if?” tanya nya padaku, dengan penuh rasa penasaran, dahinya berkerut seolah tak mengerti dngan apa yang ku tulis.

“jadi, ibarat gini, lu lagi ulangan dan lu sudah selesai ngisi semua jawabannya, terus karena lu takut ada yang belum keisi jadi lu cek ulang , begitu sama kaya if, intinya begitu, dia ngecek lagi apa semua nya sudah benar, apa maxindex sudah jadi yang terbesar”

“beneran? “

“ga tahu sih logikanya bagaimana, tapi ya... udahlah saya sudah pusing dari kemaren belum kelar kelar bahas if doang”

Next

 $temp =$Array[$maxindex];

       $Array[$maxindex]=$Array[$i];

       $Array[$i]=$temp;

    }

   return $Array;

jadi var temp menyimpan nilai array yang indexnya itu maxindex

terus array [maxindex]=array[i] jadi kan tadi maxindex sudah di hitung ya

maxindex=6,1,2,4,3,5

i = 0-6

jadi array[6]=array[0]

= array[6]=64

Jadi nilai array saat ini : $Array = [64, 34, 25, 12, 22, 11, 64];

Lalu array[i]=array[temp]

Jadi [array[0]=90, kenapa 90 ?

jadi pada baris sebelum sebelumnya temp=array[maxindex] maxindekan = 6,1,2,4,3,5

Jadi array[i]/array[0]=array[tempt]/array[6].

Hasil : $Array = [90, 34, 25, 12, 22, 11, 64];

Disinilah swap benar benar terjadi.

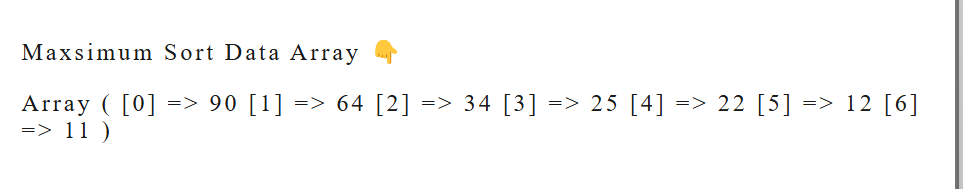
Di bawah adalah data arraynya dan contoh penggunaan dan penampilan pada browser

$data = [64,34,25,12,22,11,90];

$sortdata=maxsort($data);

echo '<br><br><br><br><p>maxsimum sort data array 👇 </p>';

print\_r($sortdata);

hasil :

**insertion array**

function insertionsort($array)  {

    $length=count($array);

bikin func baru seperti di atas lalu tentuin panjang arraynya pakai func count

for ($i=1; $i <$length; $i++) {

        $key =$array[$i];

        $j =$i - 1 ;

Lalukan perulangan seperti diatas

Penjelasan

I=1 i <length maka i tambahkan

Key(var baru yang menyimpan nilai array dan i sebagai indexnya)

Kemudian j(var baru yang menyimpan proses i-1)

Penjelasan lebih rinci :

64,34,25,12,22,11,90

Ibarat yang diatas adalah arraynya, jadi $legth=7 karena panjang array tersebut =7

Jadi selama i kurang dari 7 maka ia akan ngelooping, singkatnya i akan ngeloop 6 kali

Kemudian key = array[1] <=untuk perulangan pertama

key = array[2] <=untuk perulangan kedua

key = array[3] <=untuk perulangan ketiga

key = array[4] <=untuk perulangan ke 4

key = array[5] <=untuk perulangan ke lima

key = array[6] <=untuk perulangan KE 6

kemudian j = i-1

JADI, jika i=1 maka 1-1 =0

jika i=2 maka 2-1 =1

jika i=3 maka 3-1 =2

jika i=4 maka 4-1 =3

jika i=5 maka 5-1 =4

jika i=6 maka 6-1 =5

kemudian buat sebuah perulangan

 while($j >= 0 && $array[$j]>$key){

            $array[$j +1] = $array[$j];

            $j--;

        }

        $array[$j + 1 ] =$key;

j>= 0=>true karena j=0

**$array[$j] > $key => $array[0] = 64 > 34 => true**

Kedua kondisi true jadi loop dilanjutkan ke baris selanjutnya.

 $array[$j +1]

Array[j+1] =array[0+1]=1

$array[$j];

Array[0]

Jadi $array[$j +1] = $array[$j];

= Array[j+1] =array[0+1]= Array[0] // 0

Mka array akan menjadi seperti

Before : [64,34,25,12,22,11,90]

After : [64, 64, 25, 12, 22, 11, 90];

Karena array[1]=arraya[0] jadi di array 1 yang harusnya 34 menjadi 64 karena array 1 = array 0.

Lalu kita kurangi j--

 $j--;

Before : j= 0

After : j= -1

Maka kondisi jadi false karena j kurang dari 0, loop pun berhenti. Dan lanut ke iterasi selanjutnya.

Yaitu : $array[$j + 1] = $key;

        $array[$j + 1 ] =$key;

J= -1 jadi, j + 1 ( -1 +1) = 0 =. J = 0 maka array[0]=array[1]

Jadi :

Before [64, 64, 25, 12, 22, 11, 90];

After : [34, 64, 25, 12, 22, 11, 90];

Karena di indeks pertama pada array = 34 jadi jika array[0]= array[1] maka di indeks ke 0 = indeks ke 1.

Selesai sudah iterasi pertama maka lanjut ke iterasi ke 2.

Yaitu tepat pada baris :

        $array[$j + 1 ] =$key;

Kenapa ga di baris :

 $array[$j +1] = $array[$j];

            $j--;

Mulai iterasinya

Karena di kondisi pertamakan false, maka kita saat ini ada di loop luar dan perintah $array[$j + 1] = $key; yang di jalankan.

Jadi di iterasi ini proses nya itu kaya gini ini :

        $array[$j + 1 ] =$key;

Array[ j = 0 +1 = 1 ] key = array[2]=25

Kenapa j = 0 karena saat iterasi tadi j = 0 dan kenapa key= 2 karena kita mulai iterasi ke dua?

Lalu setelah itu kita bandingkan

($j >= 0 && $array[$j]>$key)

Kan si j = 1 true karena >= 0

Si array[j(1)=64] > $key(array[i(2)= 25) true karena j > key

Kemudian

 $j--;

J= 1 di kurang 1 = 0

Mka kondisi true karena >= 0

Jadi

($j >= 0 && $array[$j]>$key)

array[j(0)=34] > $key(array[i(2)= 25) true karena j > key

jadi array[1] di ubah jadi array [0]

array menjadi : [34, 64, 64, 12, 22, 11, 90]

kemudian lanjutkan ke $j - -

j=0 -1 = -1 kondisi false jadi lanjutkan ke loop luar

$array[$j + 1 ] =$key;

Array[-1+1= 0 ]= key[2 =25] jadi 25 di pindahkan ke indeks 0

Hasil : [25, 34, 64, 12, 22, 11, 90];

Dan begitu seterusnya. Sampe [11,12,22,25,34,64,90] beffitu kalau ga mau ribet pakai asort saja itu func php buat urutkan data dari yang terkecil ke yang terbesar, kalau untuk urutkan yang terbesar ke yang terkecil pakai artsort saja, kalau ingin diskeding semuanya mau itu key dan valuenya pakai sort saja.

Contoh penggunaan dan penampilan di browsor

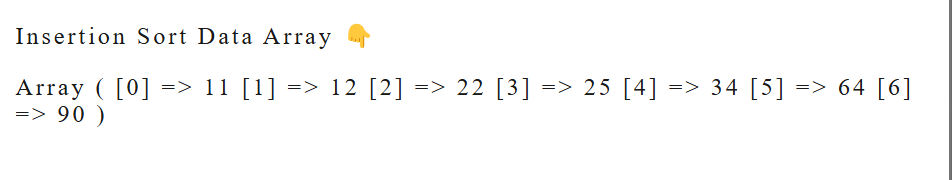
echo '<br><br><p>insertion sort data array 👇 </p>';

$sata=[64,34,25,12,22,11,90];

$sorteddata=insertionsort($sata);

print\_r($sorteddata);

hasil :



**Buble sort**

1. **Buat fun**

function bubblesort($Array){

1. Hitung panjang array

Ex :  $length =count($Array);

1. Buat perulangan

Ex :  for ($i=0; $i < $length - 1; $i++) {

       for ($j=0; $j < $length - $i - 1 ; $j++) {

i var untuk mengontrol langkahnya yang dilakukan

j untuk membandingkan nilai yang berdejatan misal 64 dan 34 begitu.

if ($Array[$j] > $Array[$j + 1]) {

jadi, misal si j = 0 maka kita membandingkan elemen pertama array = 64 dengan elment kedua array[0+1=1] array = 34 🡺 true karena 64>34

  $temp =$Array[$j];

            $Array[$j]=$Array[$j + 1];

            $Array[$j + 1 ]=$temp;

Temp menyimpan nilai dari elemet pertama untuk sementara.

Array[j] = array[j + 1 ]

Artinya taro element yang lebih kecil di posisi pertama misal 34

Jadi array[0]=array[0+1=1] jadi array 64 kan di indeks 0 dan 34 di indeks 1 maka 34 pindah jadi di indeks ke 0

Array[j+1]=tempt jadi kita taruh nilai yang sudah di simpan ke dalam temp (nilai awal array[0] = 64 pindah ke indeks 1 ]

Array

Before [64, 34, 25, 12, 22, 11, 90]

After [34, 64, 25, 12, 22, 11, 90]

Contoh penggunaan

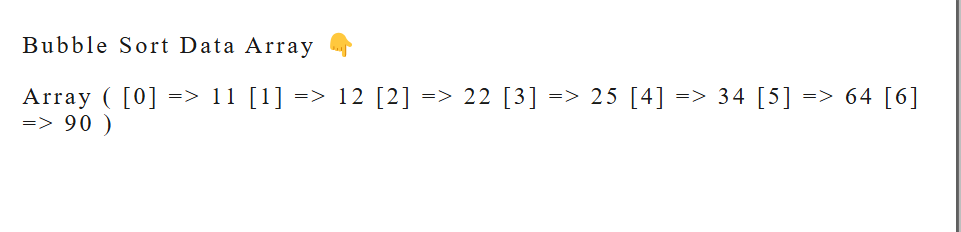
$dataa = [64,34,25,12,22,11,90];

$sortdataa=bubblesort($dataa);

echo '<br><br><p>bubble sort data array 👇 </p>';

print\_r($sortdataa);

hasil :

g 

selesai sudah!!

Ga ngerti sama penjelasanku? Kamu saja ga ngerti apalagi aku, biarlah waktu yang membuat kita bisa memahaminya.