

Daftar Tabel Tkinter

1. Daftar Widget yang Dapat Digunakan di Tkinter dengan Parameter

Widget	Deskripsi	Parameter yang Umum Digunakan
Label	Menampilkan teks atau gambar.	text, bg, fg, font, width, height, relief, padx, pady, anchor, image, justify
Button	Membuat tombol interaktif.	text, command, bg, fg, font, width, height, relief, padx, pady, state, image, borderwidth
Entry	Input teks satu baris.	textvariable, bg, fg, font, width, justify, show, state, validate, validatecommand, insert, delete
Text	Input teks multibaris.	textvariable, bg, fg, font, width, height, wrap, spacing1, spacing2, spacing3, padx, pady
Frame	Kontainer untuk widget lainnya.	bg, bd, relief, width, height, padx, pady, borderwidth, highlightbackground
Checkbutton	Pilihan dengan checkbox.	text, variable, onvalue, offvalue, bg, fg, font, width, height, command, state
Radiobutton	Pilihan dalam grup tombol radio.	text, variable, value, bg, fg, font, width, height, command, state
OptionMenu	Dropdown menu pilihan.	variable, values, bg, fg, font, width, state, command
Scale	Memilih nilai dengan slider.	from_, to, orient, variable, bg, fg, font, showvalue, length, tickinterval, resolution
Spinbox	Input teks dengan pilihan berputar.	from_, to, increment, textvariable, bg, fg, font, width, state, validate, validatecommand
Canvas	Menggambar objek (grafik, garis).	bg, height, width, bd, relief, highlightthickness, scrollregion, scrollbars, fg, font
Listbox	Menampilkan daftar item yang dapat dipilih.	listvariable, bg, fg, font, width, height, selectmode, selectbackground, selectforeground
Combobox	Dropdown menu dengan opsi input.	values, textvariable, state, width, height, postcommand, font, validate, validatecommand
PanedWindow	Membagi area aplikasi menjadi dua atau lebih bagian.	orient, bg, fg, width, height, sashwidth, sashrelief, handlewidth, handlepad, padx, pady
Scrollbar	Scrollbar untuk widget lainnya.	orient, command, width, height, bg, activebackground, sliderlength, troughcolor

Widget	Deskripsi	Parameter yang Umum Digunakan
Toplevel	Membuat jendela baru (popup).	bg, fg, width, height, title
Message	Menampilkan pesan teks multi-baris.	text, bg, fg, font, width, height, anchor, relief, padx, pady
LabelFrame	Label dengan frame di sekitarnya.	text, bg, fg, font, width, height, padx, pady, relief
PanedWindow	Membagi ruang aplikasi menjadi dua area yang dapat diubah ukurannya.	orient, bg, fg, width, height, sashwidth, sashrelief, handlewidth, handlepad, padx, pady

2. Parameter Styling Langsung Tkinter

Parameter	Deskripsi	Contoh
bg	Warna latar belakang widget	bg="#F3F2EC"
fg	Warna teks widget	fg="#E62727"
font	Jenis font dan ukurannya	font=("Arial", 12)
width	Lebar widget	width=25
height	Tinggi widget	height=5
relief	Gaya border	relief="solid"
padx	Padding horizontal	padx=10
pady	Padding vertikal	pady=5
borderwidth	Ketebalan border	borderwidth=2
highlightbackground	Warna saat widget fokus	highlightbackground="blue"

3. Parameter Sticky pada grid()

Parameter	Deskripsi	Arti
n	North	Menempel di atas
s	South	Menempel di bawah
e	East	Menempel di kanan
w	West	Menempel di kiri
ne	North-East	Sudut kanan atas
nw	North-West	Sudut kiri atas
se	South-East	Sudut kanan bawah
sw	South-West	Sudut kiri bawah

Parameter	Deskripsi	Arti
nsew	Semua arah	Mengisi seluruh sel

4. Perbedaan Layout Manager Tkinter

Metode	Cara Kerja	Kelebihan	Keterbatasan	Cocok Untuk
grid()	Menempatkan widget dalam sel baris-kolom seperti tabel.	Presisi tata letak, mudah mengatur alignment dengan <code>row/columnconfigure</code> dan <code>sticky</code> .	Perlu perencanaan struktur grid; tidak bisa dicampur dengan <code>pack()</code> dalam kontainer yang sama.	Form, dashboard, panel dengan banyak kolom/baris.
pack()	Menumpuk widget relatif terhadap sisi kontainer (top, bottom, left, right) dengan opsi fill/expand.	Cepat untuk tata letak linear; sedikit konfigurasi.	Kurang fleksibel untuk layout kompleks; kontrol posisi terbatas.	Toolbar, stack vertikal/horizontal sederhana.
place()	Menempatkan widget dengan koordinat absolut atau relatif (x, y, relx, rely).	Kontrol posisi paling detail; mendukung ukuran relatif.	Rentan terhadap perubahan ukuran window; perlu hitungan manual.	Tata letak bebas atau overlay yang membutuhkan posisi spesifik.

5. Daftar Warna dengan Hexa Desimal

Nama Warna	Hexa Desimal
Red	#FF0000
Green	#00FF00
Blue	#0000FF
White	#FFFFFF
Black	#000000
Gray	#808080
Yellow	#FFFF00
Cyan	#00FFFF
Magenta	#FF00FF
Orange	#FFA500
Purple	#800080

Nama Warna	Hexa Desimal
------------	--------------

Pink	#FFC0CB
Brown	#A52A2A
Beige	#F5F5DC
Lime	#00FF00

6. Daftar Font yang Dapat Digunakan

Nama Font	Deskripsi
-----------	-----------

Arial	Font sans-serif yang populer dan sering digunakan.
Courier	Font monospace yang sering digunakan untuk kode atau teks sejajar.
Times New Roman	Font serif klasik, sering digunakan untuk dokumen formal.
Helvetica	Font sans-serif yang lebih modern dan sering digunakan dalam desain.
Georgia	Font serif yang elegan dan mudah dibaca, sering digunakan untuk teks panjang.
Verdana	Font sans-serif yang jelas dan mudah dibaca, sering digunakan untuk web.
Tahoma	Font sans-serif yang sedikit lebih sempit dari Verdana.
Comic Sans MS	Font informal dan santai, sering digunakan dalam desain ceria.
Lucida Console	Font monospace yang digunakan untuk teks yang sejajar.
Impact	Font tebal dan mencolok, sering digunakan untuk headline.
Courier New	Font monospace klasik yang sering digunakan untuk tampilan kode.
Calibri	Font sans-serif modern yang sering digunakan di aplikasi Microsoft.
Arial Black	Font sans-serif yang tebal dan mencolok, sering digunakan untuk judul.
Trebuchet MS	Font sans-serif yang mudah dibaca dengan gaya sedikit lebih ringan.

7. Tema ttk.Style()

Nama Tema	Deskripsi	Platform
-----------	-----------	----------

clam	Tema minimalis, modern.	Semua
alt	Tema alternatif, warna terang.	Semua
default	Mengikuti OS.	Semua
classic	Tema klasik Tkinter.	Semua
vista	Gaya Windows Vista.	Windows
xpnative	Gaya Windows XP.	Windows

Nama Tema	Deskripsi	Platform
winnative	Kontrol UI Windows.	Windows

8. Fungsi Konstanta

Fungsi/Konstanta	Penjelasan	Contoh Penggunaan
tk.END	Digunakan untuk menunjukkan posisi di akhir dari widget yang memiliki teks, seperti Entry atau Text . Biasanya digunakan untuk menambahkan teks ke akhir widget.	<pre>python
entry.insert(tk.END, "Teks baru")
</pre> Menambahkan "Teks baru" di akhir teks yang sudah ada di entry .
tk.BOTH	Digunakan untuk mengatur kedua arah (horizontal dan vertikal) pada widget yang dapat digulir (scrollable), seperti Canvas , Scrollbar .	<pre>python
canvas.pack(fill=tk.BOTH, expand=True)
</pre> Memastikan bahwa widget canvas mengisi ruang baik secara horizontal maupun vertikal.
tk.X	Digunakan untuk mengatur hanya arah horizontal pada widget yang dapat digulir atau diubah ukuran, seperti Pack atau Grid .	<pre>python
entry.pack(fill=tk.X)
</pre> Widget entry akan mengisi ruang secara horizontal saja.
tk.Y	Digunakan untuk mengatur hanya arah vertikal pada widget yang dapat digulir atau diubah ukuran.	<pre>python
entry.pack(fill=tk.Y)
</pre> Widget entry akan mengisi ruang secara vertikal saja.
tk.W	Digunakan untuk meratakan widget ke arah kiri dalam konteks pengaturan grid atau pack .	<pre>python
label.grid(sticky=tk.W)
</pre> Widget label akan disesuaikan dengan sisi kiri.
tk.E	Digunakan untuk meratakan widget ke arah kanan dalam konteks pengaturan grid atau pack .	<pre>python
label.grid(sticky=tk.E)
</pre> Widget label akan disesuaikan dengan sisi kanan.
tk.N	Digunakan untuk meratakan widget ke arah atas dalam konteks pengaturan grid atau pack .	<pre>python
label.grid(sticky=tk.N)
</pre> Widget label akan disesuaikan dengan sisi atas.

Fungsi/Konstanta	Penjelasan	Contoh Penggunaan
tk.S	Digunakan untuk meratakan widget ke arah bawah dalam konteks pengaturan grid atau pack .	<code>python
label.grid(sticky=tk.S)
</code> Widget label akan disesuaikan dengan sisi bawah.
tk.CENTER	Digunakan untuk meratakan widget ke tengah dari ruang yang tersedia dalam konteks pengaturan grid atau pack .	<code>python
label.grid(sticky=tk.CENTER)
</code> Widget label akan disesuaikan dengan posisi tengah.
tk.INSERT	Digunakan untuk menunjukkan posisi kursor pada saat ini dalam widget yang memungkinkan input teks, seperti Entry atau Text .	<code>python
entry.insert(tk.INSERT, "Teks awal")
</code> Menambahkan teks "Teks awal" pada posisi kursor saat ini di entry .
tk.W (Widget)	Merujuk ke widget yang digunakan di dalam layout. Ini digunakan saat menambahkan widget dalam konteks tertentu pada widget lain.	<code>python
button.grid(row=1, column=1, padx=10, pady=10)
</code> Menentukan posisi tombol di dalam grid.
tk.TOP	Digunakan untuk menambahkan widget ke bagian atas container atau window saat menggunakan layout pack .	<code>python
frame.pack(side=tk.TOP)
</code> Widget frame akan diposisikan di bagian atas.
tk.BOTTOM	Digunakan untuk menambahkan widget ke bagian bawah container atau window saat menggunakan layout pack .	<code>python
frame.pack(side=tk.BOTTOM)
</code> Widget frame akan diposisikan di bagian bawah.
tk.LEFT	Digunakan untuk menambahkan widget ke bagian kiri container atau window saat menggunakan layout pack .	<code>python
frame.pack(side=tk.LEFT)
</code> Widget frame akan diposisikan di bagian kiri.
tk.RIGHT	Digunakan untuk menambahkan widget ke bagian kanan container atau window saat menggunakan layout pack .	<code>python
frame.pack(side=tk.RIGHT)
</code> Widget frame akan diposisikan di bagian kanan.

9. Fungsi-fungsi Dinamis dalam Python

Fungsi	Penjelasan	Contoh Penggunaan
eval()	Mengevaluasi string sebagai ekspresi Python dan mengembalikan hasilnya. Fungsi ini memungkinkan eksekusi kode dinamis. Hati-hati menggunakan eval() karena bisa mengeksekusi kode berbahaya jika input tidak terpercaya.	<pre>python
result = eval("3 + 5 * 2")
print(result) # Output: 13</pre> <p>Menilai ekspresi matematika dalam string.</p>
exec()	Fungsi ini mengeksekusi kode Python yang diberikan dalam bentuk string. Berbeda dengan eval() , exec() digunakan untuk mengeksekusi blok kode lebih kompleks, seperti definisi fungsi atau kelas.	<pre>python
code = "for i in range(3): print(i)"
exec(code) # Output: 0 1 2</pre> <p>Menjalankan loop dalam string.</p>
compile()	Mengkompilasi string menjadi objek kode Python, yang kemudian dapat dieksekusi menggunakan exec() atau dievaluasi menggunakan eval() . Berguna jika Anda ingin mengkompilasi ekspresi Python terlebih dahulu.	<pre>python
code = compile("3 + 5", "<string>", "eval")
print(eval(code)) # Output: 8</pre> <p>Mengkompilasi ekspresi Python untuk evaluasi.</p>
globals()	Mengembalikan kamus global saat ini dalam bentuk dictionary. Ini memungkinkan akses ke variabel global dari dalam fungsi atau blok kode.	<pre>python
def test():
 print(globals())
test() # Output: {'__name__': '__main__', 'test': <function test>}</pre> <p>Menampilkan variabel global.</p>
locals()	Mengembalikan kamus lokal saat ini dalam bentuk dictionary. Berguna untuk mengakses variabel dalam ruang lingkup lokal.	<pre>python
def test():
 a = 5
 print(locals())
test() # Output: {'a': 5}</pre> <p>Menampilkan variabel lokal dalam fungsi.</p>
getattr()	Digunakan untuk mendapatkan atribut dari objek menggunakan nama atribut dalam bentuk string. Ini berguna jika Anda tidak tahu nama atributnya saat menulis kode.	<pre>python
class MyClass:
 x = 10
obj = MyClass()
print(getattr(obj, "x")) # Output: 10</pre> <p>Mendapatkan atribut objek secara dinamis.</p>
setattr()	Digunakan untuk menetapkan atribut ke objek dengan nama atribut yang diberikan dalam bentuk string. Ini memungkinkan penetapan nilai atribut dinamis.	<pre>python
class MyClass:
 pass
obj = MyClass()
setattr(obj, "x", 20)
print(obj.x) # Output: 20</pre> <p>Menetapkan atribut objek secara dinamis.</p>

Fungsi	Penjelasan	Contoh Penggunaan
delattr()	Digunakan untuk menghapus atribut dari objek berdasarkan nama atribut yang diberikan dalam bentuk string.	<pre>python
class MyClass:
 x = 10
obj = MyClass()
delattr(obj, "x")
print(hasattr(obj, "x")) # Output: False</pre> <p>Menghapus atribut dari objek secara dinamis.</p>
callable()	Memeriksa apakah objek dapat dipanggil atau tidak (misalnya, fungsi, metode, atau objek yang mengimplementasikan metode <code>__call__</code>).	<pre>python
def test():
 pass
print(callable(test)) # Output: True</pre> <p>Memeriksa apakah objek dapat dipanggil sebagai fungsi.</p>
repr()	Mengembalikan representasi string dari objek yang dapat diekspresikan dalam bentuk kode Python yang valid. Berguna untuk debugging dan logging.	<pre>python
x = [1, 2, 3]
print(repr(x)) # Output: [1, 2, 3]</pre> <p>Menampilkan representasi string dari objek.</p>
eval() (Alternatif)	Jika Anda membutuhkan alternatif lebih aman daripada <code>eval()</code> , Anda bisa menggunakan <code>ast.literal_eval()</code> untuk menilai ekspresi yang hanya mengandung nilai literal (misalnya, angka, string, tuple, list).	<pre>python
import ast
print(ast.literal_eval([1, 2, 3])) # Output: [1, 2, 3]</pre> <p>Evaluasi ekspresi literal secara aman.</p>