

# TradingView — “Loading terus / Muncul sebentar lalu hilang” (Replit/Next.js)

Berikut **root cause** paling umum + **patch** siap tempel supaya chart stabil di Replit Preview.

---

## 1) Penyebab Paling Umum

1. **Script loader duplikat / race**
  2. `tv.js` di-append berkali-kali → widget sempat muncul lalu dibuang/overwritten.
  3. Memakai `existingScript.remove()` memicu race condition.
  4. **Strict Mode (Dev) → double effect**
  5. Next.js/React dev mode memanggil `useEffect` dua kali → inisialisasi & cleanup berulang → flicker/hilang.
  6. **Unmount saat state berubah**
  7. Komponen parent (tabs/conditional render) meng-unmount `<TradingViewWidget/>` ketika `data` / `tab` berubah → widget hilang.
  8. **CSP / Sandbox Iframe Replit**
  9. `s3.tradingview.com` diblokir oleh Content-Security-Policy/sandbox → script tidak bisa jalan konsisten.
  10. **Container collapse**
  11. Tinggi elemen induk berubah (misal `display:none` saat tab pindah) → TV tidak redraw dan terlihat “hilang”.
- 

## 2) Checklist Diagnosis Cepat

- **Network tab:** pastikan `https://s3.tradingview.com/tv.js` **status 200**.
  - **Console:** tidak ada error `TradingView is not defined` / `Blocked by CSP`.
  - **Elements:** ada `<iframe>` TradingView di dalam container (bukan kosong).
  - **Styles:** container **tetap** `height: 500px` (atau `min-height`) saat tab berganti.
  - **CSP:** izinkan domain `s3.tradingview.com` pada `script-src` & `frame-src` (`https://s3.tradingview.com https://*.tradingview.com`).
-

### 3) Patch Wajib (Idempotent Loader + Guard Strict Mode)

**A. Loader tunggal** (jangan pernah `remove()` script):

```
// singleton promise – hindari duplikat loader & race
if (!window.__tvScriptPromise__) {
  window.__tvScriptPromise__ = new Promise<void>((resolve, reject) => {
    const exist = document.getElementById('tradingview-widget-script') as
    HTMLScriptElement | null;
    if (exist) {
      exist.addEventListener('load', () => resolve());
      exist.addEventListener('error', () => reject(new Error('TV load
error')));
      return;
    }
    const s = document.createElement('script');
    s.id = 'tradingview-widget-script';
    s.src = 'https://s3.tradingview.com/tv.js';
    s.async = true;
    s.onload = () => resolve();
    s.onerror = () => reject(new Error('TV load error'));
    document.head.appendChild(s);
  });
}
await window.__tvScriptPromise__;
```

**B. StrictMode guard** (hindari inisialisasi ganda di dev):

```
const didInit = useRef(false);
useEffect(() => {
  if (didInit.current) return; // cegah run 2x
  didInit.current = true;
  initWidget();
}, [initWidget]);
```

**C. Jangan unmount saat state kosong**

Render `<TradingViewWidget/>` **selalu**, jangan bergantung pada `data`. Tampilkan skeleton di atasnya saja.

**D. Re-init hanya saat perlu**

Dependensi `initWidget`: `tvSymbol`, `interval`, `theme`, `studies`. **Bukan** `data`.

**E. Container stabil**

Pastikan parent/tab tidak pakai `display:none` saat TV butuh render pertama. Gunakan `visibility:hidden` dulu lalu `visible` setelah ready.

## 4) Contoh Komponen Final (intisari)

Versi lengkap sudah kubuat di canvas **"TradingViewWidget—SSR-safe, Idempotent Loader, Symbol Switch."** Ini cuplikan kuncinya.

```
const isBrowser = typeof window !== 'undefined';

function loadTv(): Promise<void> {
  if (!isBrowser) return Promise.resolve();
  if (window.TradingView?.widget) return Promise.resolve();
  if (window.__tvScriptPromise__) return window.__tvScriptPromise__!;
  // ... (kode loader A di atas)
}

export function TradingViewWidget({ tvSymbol = 'OKX:SOLUSDTPERP', interval =
'1H', theme = 'dark' }) {
  const ref = useRef<HTMLDivElement | null>(null);
  const didInit = useRef(false);

  const initWidget = useCallback(async () => {
    await loadTv();
    if (!ref.current || !window.TradingView?.widget) return;
    ref.current.innerHTML = '';
    const mount = document.createElement('div');
    mount.id = `tv_${Date.now()}`;
    mount.style.height = '500px';
    ref.current.appendChild(mount);

    new window.TradingView.widget({
      autosize: true,
      symbol: tvSymbol,
      interval: interval === '1H' ? '60' : interval === '4H' ? '240' :
interval,
      theme,
      container_id: mount.id,
      studies: ['Volume', 'RSI@tv-basicstudies', 'MACD@tv-basicstudies'],
      // overrides ...
    });
  }, [tvSymbol, interval, theme]);

  useEffect(() => {
    if (didInit.current) return; // guard StrictMode
    didInit.current = true;
    initWidget();
  }, [initWidget]);

  return <div ref={ref} className="h-[500px] w-full bg-gray-900 rounded-lg" /
>;
}
```

---

## 5) Replit / CSP Setup

Jika pakai header/meta CSP, pastikan:

```
script-src 'self' https://s3.tradingview.com ...  
frame-src https://*.tradingview.com ...  
connect-src https://*.tradingview.com https://s3.tradingview.com ...
```

Dan **matikan extension ad-blocker** di Replit Preview/tab baru.

---

## 6) Quick Fix Flow (5 menit)

1) Ganti komponen dengan versi canvas (atau tempel Patch A-E). 2) Pastikan `<TradingViewWidget/>` **selalu dirender** (jangan conditional `data &&` untuk mount). 3) Reload Replit Preview **di tab baru** (bukan iframe kecil) untuk hindari sandbox. 4) Cek Network: `tv.js` = 200; Console = kosong dari error CSP. 5) Jika masih flicker: nonaktifkan `React.StrictMode` sementara di dev untuk konfirmasi root cause (bila hilang → berarti double effect; pakai guard ref di atas).

---

## 7) Bonus: Symbol Switch (OKX ↔ Binance)

Set `tvSymbol` ke `OKX:SOLUSDTPERP` atau `BINANCE:SOLUSDT`. Render ulang **tanpa** unmount parent.

---

Kalau kamu mau, aku bisa langsung commitkan file widget versi stabil dan menambahkan CSP hint + toggle visibility agar tab switching tidak memaksa redraw TV.