

# REFERENSI CEPAT NPM

Manajemen paket, skrip, versioning, publishing, workspaces

## Instalasi

### Menginstal npm & Node

```
node -v && npm -v # periksa versi terpasang
npm install -g npm@latest # perbarui npm itu sendiri
nvm install --lts # install Node LTS via nvm
nvm use 20 # beralih ke Node 20
```

### Perintah Install

```
npm install # Instal semua dependensi dari package.json
npm install pkg # Tambahkan paket sebagai dependency
npm install -D pkg # Tambahkan paket sebagai devDependency
npm install -g pkg # Instal paket secara global
npm install pkg@2.1.0 # Instal versi tertentu
npm ci # Instal bersih dari lock file (CI/CD)
npm uninstall pkg # Hapus paket
```

## Manajemen Paket

### Mengelola Paket

```
npm ls # daftar paket terpasang
npm ls --depth=0 # hanya level teratas
npm outdated # periksa versi terbaru
npm update # perbarui dalam rentang semver
npm audit # periksa kerentanan
```

### Perintah Manajemen

```
npm ls # Daftar paket terpasang sebagai pohon
npm outdated # Tampilkan paket yang ada versi lebih baru
npm update [pkg] # Perbarui paket dalam rentang semver
npm audit # Audit dependensi untuk kerentanan
npm audit fix # Perbaiki otomatis dependensi rentan
npm prune # Hapus paket yang tidak diperlukan
npm dedupe # Ratakan pohon dependensi untuk mengurangi duplikasi
```

## Skrip

### Menjalankan Skrip

```
npm run build # jalankan skrip "build"
npm test # pintasan untuk skrip "test"
npm start # pintasan untuk skrip "start"
npm run lint -- --fix # teruskan argumen ke skrip
npm run dev & # jalankan di latar belakang
```

### Siklus Hidup Skrip

```
npm test / npm t # Jalankan `scripts.test`
npm start # Jalankan `scripts.start`
npm run <name> # Jalankan skrip kustom apapun
pre<name> # Berjalan otomatis sebelum <name>
post<name> # Berjalan otomatis setelah <name>
npm run # Daftar semua skrip tersedia
```

## package.json

### Inisialisasi & Field

```
npm init # setup interaktif
npm init -y # terima semua default
npm pkg set name="my-app" # set field
npm pkg get version # baca field
```

### Field Utama

```
name # Nama paket (huruf kecil, tanpa spasi)
version # Versi saat ini (semver: major.minor.patch)
main # Entry point untuk CommonJS (require)
module # Entry point untuk ES modules (bundler)
type # "module" untuk ESM, "commonjs" untuk CJS (default)
scripts # Perintah bernama (build, test, start, dll.)
dependencies # Dependensi produksi
devDependencies # Dependensi hanya untuk pengembangan
engines # Rentang versi Node/npm yang diperlukan
```

## Versioning

### Perintah Versi

```
npm version patch # 1.0.0 -> 1.0.1
npm version minor # 1.0.1 -> 1.1.0
npm version major # 1.1.0 -> 2.0.0
npm version 3.2.1 # set versi eksplisit
npm version prerelease --preid=beta # 1.0.0-beta.0
```

### Rentang Semver

```
^1.2.3 # Kompatibel: >=1.2.3 <2.0.0 (default)
~1.2.3 # Level patch: >=1.2.3 <1.3.0
1.2.3 # Hanya versi tepat
>=1.0.0 <2.0.0 # Rentang eksplisit
* # Versi apa saja
1.x / 1.2.x # Rentang wildcard
latest # Tag versi terbaru yang dipublikasikan
```

## Publishing

### Alur Kerja Publish

```
npm login # autentikasi ke registry
npm publish # publish paket publik
npm publish --access public # paket scoped sebagai publik
npm unpublish pkg@1.0.0 # hapus versi tertentu
npm deprecate pkg@<2 "Use v2+" # tandai versi lama usang
```

### Referensi Publish

```
npm login # Autentikasi dengan registry npm
npm publish # Publish paket ke registry
npm pack # Buat tarball tanpa publishing
npm unpublish # Hapus versi yang dipublikasikan (dalam 72 jam)
npm deprecate # Tandai versi sebagai usang
npmignore # File yang dikucualikan dari paket yang dipublikasikan
```

**files (package.json)** Allowlist file yang disertakan dalam paket

## Workspaces

### Perintah Workspace

```
npm init -w packages/core # buat workspace
npm install -w packages/core lodash # install di workspace
npm run build -w workspaces # jalankan di semua workspace
npm run test -w packages/api # jalankan di workspace tertentu
npm ls -w workspaces # daftar dependensi workspace
```

### Konfigurasi Workspace

```
workspaces (package.json) # Array glob workspace: ["packages/**"]
-w / --workspace # Target workspace tertentu
--workspaces # Jalankan perintah di semua workspace
--include-workspace-root # Sertakan paket root dalam operasi workspace
npm install (root) # Instal semua dependensi workspace
Hoisting # Dependensi bersama diangkat ke root node_modules
```

## npm

### Menjalankan dengan npx

```
npx create-react-app my-app # jalankan tanpa menginstal
npx tsc --init # jalankan bin lokal atau remote
npx -p typescript tsc file.ts # tentukan paket secara eksplisit
npx --yes create-next-app # lewat prompt install
npx node@18 -e "console.log('hi')" # jalankan dengan Node tertentu
```

### Opsi npx

```
npx cmd # Jalankan cmd dari node_modules/bin lokal atau remote
npx -p pkg cmd # Instal pkg, lalu jalankan cmd
npx --yes cmd # Konfirmasi otomatis prompt instalasi
npx --no cmd # Tolak instalasi — gagal jika tidak ada lokal
npx -c 'cmd' # Jalankan perintah shell dengan PATH npx
npx node@ver # Jalankan versi Node.js tertentu
```

## Konfigurasi

### Perintah Config

```
npm config list # tampilkan konfigurasi saat ini
npm config set registry https://r.npmjs.com/
npm config set init-author-name "Name"
npm config get prefix # path install global
npm config delete key # hapus nilai konfigurasi
```

### Referensi Config

```
npmrc (project) # File konfigurasi per-proyek
./ .npmrc # File konfigurasi per-pengguna
registry # URL registry paket
save-exact # true untuk pin versi tepat saat install
engine-strict # true untuk menerapkan field engines
fund # false untuk menyembunyikan pesan funding
audit # false untuk lewat audit saat install
```

## Pola Umum

### One-Liner

```
npm ls --depth=0 --json | jq '.dependencies | keys[]'
npm outdated --long # tampilkan tipe dan homepage
npm cache clean --force # bersihkan cache npm
npm explain pkg # kenapa pkg terpasang?
npm exec -- envinfo --system # info sistem untuk laporan bug
```

### Resep

```
Hanya lock file # `npm ci` — instal bersih dari package-lock.json
Periksa lisensi # `npm license-checker --summary`
Temukan dep tak terpakai # `npm depcheck`
Ukuran bundle # `npx bundlephobia-cli pkg` — periksa ukuran paket
Upgrade semua # `npm npm-check-updates -u && npm install`
Registry lokal # `npx verdaccio` — jalankan registry privat
```