

Versioning

Versioning untuk kode PHP mengacu pada praktik memberikan versi numerik atau label tertentu pada setiap rilis atau perubahan signifikan dalam kode PHP Anda. Ini penting untuk mengelola dan melacak evolusi kode, memfasilitasi manajemen perubahan, pemeliharaan, dan kolaborasi tim. Berikut adalah beberapa praktik umum dalam versioning untuk kode PHP:

1. Penomoran Versi

SemVer (Semantic Versioning): Menggunakan format tiga angka: `MAJOR.MINOR.PATCH`.

- **MAJOR:** Versi utama, berubah ketika ada perubahan yang tidak kompatibel ke belakang.
- **MINOR:** Versi minor, berubah ketika menambah fitur secara kompatibel ke belakang.
- **PATCH:** Versi perbaikan, berubah untuk perbaikan bug secara kompatibel ke belakang.

Contoh: `1.0.0`, `1.2.3`, `2.1.0`.

2. Pengelolaan Kode Sumber

- **Git:** Gunakan Git untuk melacak setiap perubahan kode. Setiap kali Anda melakukan commit, berikan pesan yang jelas tentang perubahan yang dilakukan.

```
git commit -m "Menambahkan fitur pembayaran dengan kartu kredit"
```

- **Branching:** Gunakan branching untuk mengembangkan fitur baru atau memperbaiki bug tanpa mempengaruhi kode utama yang stabil.

```
git checkout -b fitur-pembayaran
```

3. Manajemen Repository

- **Tagging:** Gunakan tag Git untuk menandai rilis tertentu yang sudah stabil.

```
git tag -a v1.0.0 -m "Versi 1.0.0"  
git push origin v1.0.0
```

- **Release Notes:** Buat catatan rilis (release notes) yang menjelaskan perubahan utama dan perbaikan yang ada dalam setiap versi.

4. Penggunaan Manajer Paket

- **Composer:** Jika menggunakan dependensi PHP, spesifikasikan versi yang diinginkan dalam `composer.json`.

```
{
  "require": {
    "vendor/package": "1.2.3"
  }
}
```

5. CI/CD dan Otomatisasi

- **Continuous Integration (CI):** Integrasikan alat CI seperti Jenkins, GitLab CI, atau GitHub Actions untuk mengotomatisasi build, pengujian, dan deploy setiap kali ada perubahan dalam kode.
- **Continuous Deployment (CD):** Gunakan alat CD untuk mengotomatisasi proses deploy ke lingkungan produksi setelah berhasil melewati pengujian.

Dengan menerapkan praktik versioning ini, Anda dapat mengelola dan melacak evolusi kode PHP dengan lebih terstruktur dan terorganisir, memungkinkan kolaborasi yang lebih efisien antar tim pengembang dan meminimalkan risiko perubahan yang tidak diinginkan.

Revision #1

Created 13 December 2024 13:39:00 by Admin

Updated 13 December 2024 14:01:57 by Admin