Rekayasa perangkat lunak

Dari Wikipedia bahasa Indonesia, ensiklopedia bebas

**Rekayasa perangkat lunak** (RPL, atau dalam [bahasa Inggris](https://id.wikipedia.org/wiki/Bahasa_Inggris): *Software Engineering* atau *SE*) adalah satu bidang profesi yang mendalami cara-cara pengembangan [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) termasuk pembuatan, pemeliharaan, manajemen organisasi pengembanganan perangkat lunak dan manajemen kualitas.

*IEEE Computer Society* mendefinisikan rekayasa perangkat lunak sebagai penerapan suatu pendekatan yang sistematis, disiplin dan terkuantifikasi atas pengembangan, penggunaan dan pemeliharaan perangkat lunak, serta studi atas pendekatan-pendekatan ini, yaitu penerapan pendekatan **engineering** atas perangkat lunak.[[1]](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#cite_note-BoDu04-1)

rekayasa [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) adalah pengubahan perangkat lunak itu sendiri guna mengembangkan, memelihara, dan membangun kembali dengan menggunakan prinsip reakayasa untuk menghasilkan perangkat lunak yang dapat bekerja lebih efisien dan efektif untuk pengguna.

kriteria yang dapat digunakan sebagai acuan dalam merekayasa perangkat lunak:

1. dapat terus dirawat dan dipelihara(maintainability)

2. dapat mengikuti perkembangan teknologi(dependability)

3. dapat mengikuti keinginan pengguna(robust)

4. efektif dan efisien dalam menggunakan energi dan penggunaannya

5. dapat memenuhi kebutuhan yang diinginkan(usability)[[2]](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#cite_note-2)

**Daftar isi**

  [[sembunyikan](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak)]

* [1Sejarah](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#Sejarah)
* [2Di Indonesia](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#Di_Indonesia)
* [3Subdisiplin](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#Subdisiplin)
* [4Referensi](https://id.wikipedia.org/wiki/Rekayasa_perangkat_lunak#Referensi)

Sejarah[[sunting](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&veaction=edit&vesection=1) | [sunting sumber](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&action=edit&section=1)]

*!Artikel utama untuk bagian ini adalah:*[*Sejarah rekayasa perangkat lunak*](https://id.wikipedia.org/wiki/Sejarah_rekayasa_perangkat_lunak)

Istilah *software engineering*, pertama kali digunakan pada akhir tahun 1950-an dan sekitar awal 1960-an. Pada tahun 1968, [NATO](https://id.wikipedia.org/wiki/NATO) menyelenggarakan konferensi tentang*software engineering* di Jerman dan kemudian dilanjutkan pada tahun 1969. Meski penggunaan kata *software engineering* masukan konferensi tersebut menimbulkan debat tajam tentang aspek *engineering* dari pengembangan [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak), banyak pihak yang menganggap konferensi tersebutlah yang menjadi awal tumbuhnya profesi **rekayasa perangkat lunak**

Di Indonesia[[sunting](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&veaction=edit&vesection=2) | [sunting sumber](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&action=edit&section=2)]

Rekayasa Perangkat Lunak Di indonesia dijadikan disiplin ilmu yang dipelajari mulai tingkat [Sekolah Menengah Kejuruan](https://id.wikipedia.org/wiki/Sekolah_Menengah_Kejuruan) sampai tingkatan [Perguruan Tinggi](https://id.wikipedia.org/wiki/Perguruan_Tinggi). Di tingkat SMK, jurusan ini sudah memiliki kurikulum materi pelajaran sendiri yang sudah ditentukan oleh Dinas Pendidikan. Rekayasa Perangkat Lunak Di tingkat SMK biasanya mempelajari materi materi seperti [Bahasa Pemrograman](https://id.wikipedia.org/wiki/Bahasa_Pemrograman), [Desain Web](https://id.wikipedia.org/w/index.php?title=Desain_Web&action=edit&redlink=1), Pengetahuan tetang Undang Undang HAKI dan ITE, dan sebagainya, tergantung dari sekolah dan kurikulum tiap tahunnya.

Subdisiplin[[sunting](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&veaction=edit&vesection=3) | [sunting sumber](https://id.wikipedia.org/w/index.php?title=Rekayasa_perangkat_lunak&action=edit&section=3)]

* [Kualitas perangkat lunak](https://id.wikipedia.org/w/index.php?title=Kualitas_perangkat_lunak&action=edit&redlink=1) (*software quality*)
* [Teknik kebutuhan perangkat lunak](https://id.wikipedia.org/wiki/Teknik_kebutuhan_perangkat_lunak) (*requirements engineering*)
* [Manajemen proyek perangkat lunak](https://id.wikipedia.org/w/index.php?title=Manajemen_proyek_perangkat_lunak&action=edit&redlink=1) (project management)
* [Pengujian perangkat lunak](https://id.wikipedia.org/wiki/Pengujian_perangkat_lunak) (*software testing)*
* [Metode formal](https://id.wikipedia.org/w/index.php?title=Metode_formal&action=edit&redlink=1) (*formal methods*)
* [Penjaminan mutu perangkat lunak](https://id.wikipedia.org/w/index.php?title=Penjaminan_mutu_perangkat_lunak&action=edit&redlink=1) (*software quality assurance*)
* [Verifikasi perangkat lunak](https://id.wikipedia.org/w/index.php?title=Verifikasi_perangkat_lunak&action=edit&redlink=1) (*software verification*)
* [Perangkat lunak terdistribusi](https://id.wikipedia.org/w/index.php?title=Perangkat_lunak_terdistribusi&action=edit&redlink=1) (*distributed software engineering*)
* [Manajemen konfigurasi perangkat lunak](https://id.wikipedia.org/w/index.php?title=Manajemen_konfigurasi_perangkat_lunak&action=edit&redlink=1) (*software configuration management*)
* [Perawatan perangkat lunak](https://id.wikipedia.org/w/index.php?title=Perawatan_perangkat_lunak&action=edit&redlink=1) (*software maintenance*)
* [permodelan dan perancangan perangkat lunak](https://id.wikipedia.org/w/index.php?title=Perancangan_perangkat_lunak&action=edit&redlink=1) (*software design and modelling*)
* [Arsitektur perangkat lunak](https://id.wikipedia.org/wiki/Arsitektur_perangkat_lunak) (*software architectures*)
* [Arsitektur perangkat lunaks](https://id.wikipedia.org/w/index.php?title=Arsitektur_perangkat_lunaks&action=edit&redlink=1) (*software architecturess*)