



# Pemrograman Rust

Bambang Purnomosidi D. P. <bambangpdp@gmail.com>

Version 0.0.1, 2020-06-02

# Daftar Isi

Pengantar .....	1
Penghargaan .....	1
Bagian I: Pengenalan dan Ekosistem .....	2
1. Tentang Rust .....	3
2. Instalasi Rust .....	4
2.1. Rilis Rust .....	4
2.2. Rust Playground .....	4
3. IDE untuk Rust .....	6
4. Ekosistem Rust .....	7
Bagian II: Sintaksis dan Semantik dari Rust .....	8
5. Variabel dan Komentar .....	9
6. Tipe Data .....	10
7. Fungsi .....	11
8. Pengandali Aliran Program .....	12
9. Struktur Data .....	13
10. Ownership .....	14
11. Enum .....	15
12. Perbandingan Pola .....	16
Bagian III: Pustaka Standar Rust .....	17
13. Mengelola String .....	18
14. Pemrograman Konkuren .....	19
15. Net .....	20
16. Path .....	21
17. Sistem File .....	22
18. Operasi I/O .....	23
19. Mengelola Proses .....	24
Bagian IV: Topik Khusus .....	25
20. Akses Basis Data .....	26
21. Pemrograman Web .....	27
22. Antarmuka Teks .....	28
23. Antarmuka Grafis .....	29

# Pengantar

Sebuah buku tentang [Bahasa Pemrograman Rust](#).

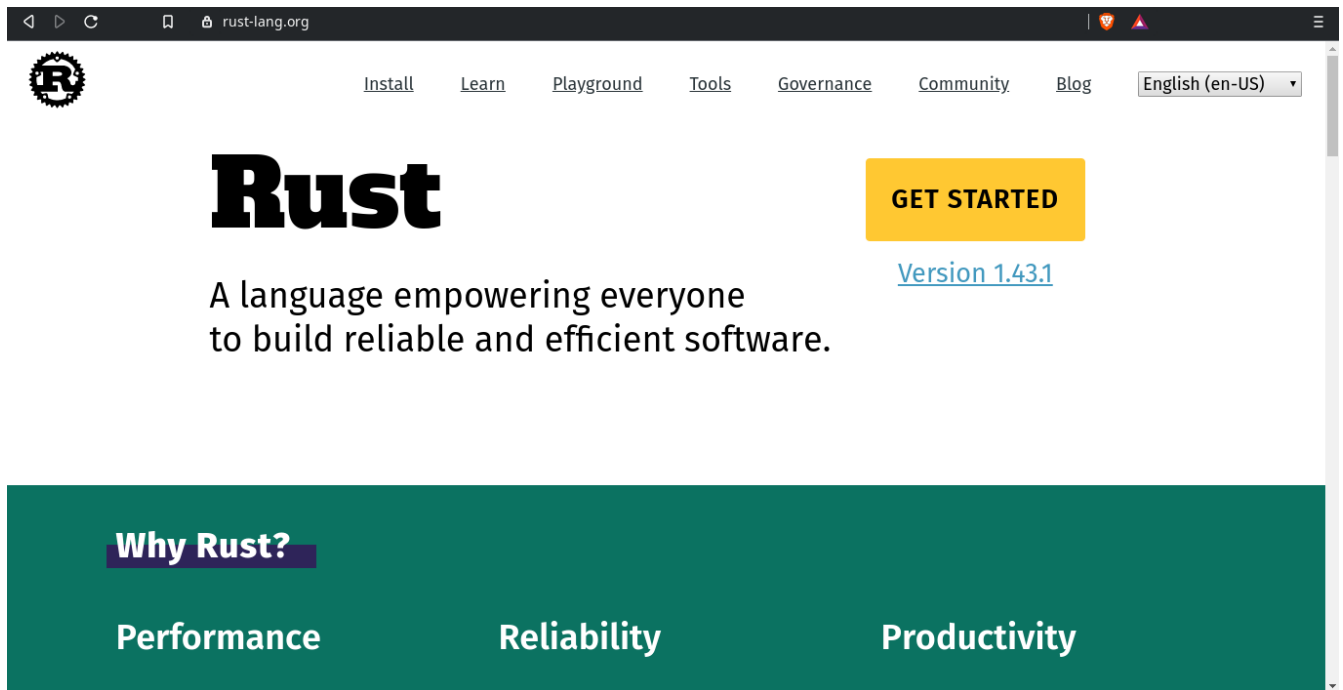
## Penghargaan

Kami mengucapkan terima kasih atas partisipasi dari rekan-rekan semua.

# Bagian I: Pengenalan dan Ekosistem

Bagian ini menjelaskan tentang gambaran umum dari Rust serta persiapan untuk membangun aplikasi menggunakan Rust

# Bab 1. Tentang Rust



# Bab 2. Instalasi Rust

Untuk menggunakan Rust, tentu saja anda harus melakukan instalasi terhadap Rust dan ekosistem yang bisa digunakan untuk mendukung proses membangun software menggunakan Rust. Bab ini membahas tentang berbagai cara yang bisa digunakan untuk mulai menggunakan Rust.

## 2.1. Rilis Rust

Rust mempunyai 3 kategori rilis:

1. *Stable*: rilis stabil, dengan *test* yang dilakukan secara menyeluruh.
2. *Beta*: rilis versi ini merupakan rilis yang disiapkan untuk menjadi versi stabil berikutnya..
3. *Nightly*: rilis versi ini merupakan rilis yang berisi berbagai eksperimen yang mungkin bisa masuk ke versi stabil berikutnya (setelah melalui versi *Beta*), maupun tidak akan pernah dimasukkan ke rilis Rust.

Saat membangun aplikasi, pemrogram bebas untuk menggunakan kategori rilis manapun. Meskipun demikian, dianjurkan untuk menggunakan versi *Stable* karena fitur yang ada di dalamnya adalah fitur-fitur yang sudah stabil sehingga memudahkan pemrogram untuk *maintain* aplikasi yang dikembangkan.

Untuk semua rilis tersebut, Rust menggunakan pedoman yang disebut dengan **Semantic Versioning**. Dengan menggunakan pedoman ini, setiap rilis Rust terdiri atas 3 bagian:

1. MAJOR: rilis dengan perubahan API (*Application Programming Interface*) yang tidak kompatibel dengan versi MAJOR sebelumnya.
2. MINOR: rilis dengan penambahan fungsionalitas yang kompatibel dengan versi sebelumnya.
3. PATCH: rilis dengan perbaikan terhadap *bugs* yang kompatibel dengan versi sebelumnya.

Sebagai contoh, versi 1.44.0 dari versi Rust berisi Rust dengan semua API yang kompatibel dengan versi 1.x.x sebelumnya. Angka 44 berarti penambahan fungsionalitas yang bersifat kompatibel dengan versi penambahan fungsionalitas sebelumnya. Angka 0 berarti sama sekali belum ada perubahan perbaikan *bug* untuk versi 1.44 tersebut.

## 2.2. Rust Playground

Jika kebutuhan kita hanya untuk mencoba beberapa bagian kode sumber, maka kita cukup hanya menggunakan **Rust Playground** saja. Setelah mengakses URL tersebut, kita bisa menuliskan kode sumber dan menjalankan kode sumber tersebut tanpa perlu melakukan instalasi peranti pengembangan Rust.

play.rust-lang.org

RUN ▶

...

DEBUG ▾

STABLE ▾

...

SHARE

TOOLS ▾

CONFIG ▾

?

1 fn main() {  
2 println!("Hello, world!");  
3 }

Execution

Close

Standard Error

Compiling playground v0.0.1 (/playground)  
Finished dev [unoptimized + debuginfo] target(s) in 0.70s  
Running `target/debug/playground`

Standard Output

Hello, world!

# Bab 3. IDE untuk Rust

IDE Rust



# Bab 4. Ekosistem Rust

```
use ferris_says::say;                                ①
use std::io::{stdout, BufWriter};

fn main() {
    let stdout = stdout();
    let message = String::from("Hello fellow Rustaceans!");
    let width = message.chars().count();

    let mut writer = BufWriter::new(stdout.lock());
    say(message.as_bytes(), width, &mut writer).unwrap(); ②
}
```

① Penjelasan 1

② Penjelasan 2

# Bagian II: Sintaksis dan Semantik dari Rust

Bagian ini menjelaskan tentang sintaksis dari bahasa pemrograman Rust. Beberapa bagian sudah membahas tentang pustaka standar sesuai pada materi pembahasan. Jika merupakan pustaka standar, bagian tersebut akan diberi catatan.

# Bab 5. Variabel dan Komentar

Var dan komentar

# Bab 6. Tipe Data

Tipe data

# Bab 7. Fungsi

Fungsi

# Bab 8. Pengandali Aliran Program

Pengendali aliran program

# Bab 9. Struktur Data

Struktur data

# Bab 10. Ownership

Ownership



# Bab 11. Enum

Enum

# Bab 12. Pembandingan Pola

Pembandingan pola

# Bagian III: Pustaka Standar Rust

Bagian ini menjelaskan berbagai pustaka standar dari Rust. Pustaka standar merupakan API yang menjadi bagian dari Rust dan bisa diakses setelah kita melakukan instalasi Rust tanpa perlu melakukan instalasi tambahan lain. Bagian ini tidak menjelaskan secara rinci semua pustaka standar, tetapi hanya beberapa saja. Setelah itu, pembaca bisa melihat pada dokumentasi lengkap dari pustaka standar dari Rust untuk mengetahui lebih lanjut.

## Bab 13. Mengelola String

## **Bab 14. Pemrograman Konkuren**

## Bab 15. Net

## Bab 16. Path

## Bab 17. Sistem File



## Bab 18. Operasi I/O

# Bab 19. Mengelola Proses

# Bagian IV: Topik Khusus

Bagian ini menjelaskan berbagai topik pemrograman khusus yang bisa dilakukan dengan menggunakan Rust.

## Bab 20. Akses Basis Data

# Bab 21. Pemrograman Web

## Bab 22. Antarmuka Teks

## Bab 23. Antarmuka Grafis