hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

Ruby for Beginner

A Programmer's Best Friend

Speaker:
RIZQI NUR ASSYAUFI

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

RIZQI NUR ASSYAUFI

bandithijo.github.io

bandithijo bandithijo gmail.com

t.me

/GNULinuxIndonesia /VimID
/JekyllID /ArchLinuxID
/dotfiles_id /ruby_id

Ruby for Beginner

A Programmer's Best Friend

bandithijo

- 2017
  Agung Setiawan
  IDRails.com
  Online Course
- 2019
  VirtualSpirit Technology
  Junior Backend Ruby on Rails Developer

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Introduction to Ruby

Ruby for Beginner
A Programmer's Best Friend

> Sedikit tentang Ruby

```
:creator   ⇒ Yukihiro “Matz” Matsumoto
:inspired ⇒ [ Perl, Smalltalk, Eiffel, Ada, Lisp ]
:public_release   ⇒ 1995
:mass_acceptance ⇒ 2006 < DHH.create(Ruby on Rails,
                                     Basecamp,
                                     2003)
```

```
Matz.said(“trying to make Ruby natural, not simple”)

   _.append(“Ruby is simple in appearance,
             but is very complex inside,
             just like our human body.”)
```

Source: About Ruby - http://www.ruby-lang.org/en/about/

> Is Ruby Dead Programming Language?

:dev_a ⇒ “We moved away from Ruby.
It’s a dying language...
and it has scalability problem”

:dev_b ⇒ “Let’s host our projects on **GitHub** & **GitLab**, use **Stripe** as our payment processor, crowdfund with **KickStarter**, and livestream on **Twitch**. Also, write about our industry and programming on **dev.to**”

Source: Nate Hopkins @hopsoft - https://twitter.com/hopsoft/status/1169706322851328000

> Is Ruby Dead Programming Language?

```
success_stories("Ruby")
⇒ {
Simulations: [Nasa, Motorola, etc.]
3D_Modeling: [Google Scketchup, etc.]
Robotics   : [Morpha Project (Siemens), etc.]
Security   : [Metasploit Framework, etc.]
Web Apps   : [Basecamp, etc.]
etc.       : [...] }
```

Source: Ruby Lang Success Stories - http://www.ruby-lang.org/en/documentation/success-stories/

> Is Ruby Dead Programming Language?

```
Ruby.who("INDONESIA").value
```

```
⇒ [ Bukalapak, Cookpad Indonesia, Midtrans,
  Quipper, Vidio, HIJUP, Alodokter, Apisentris,
  BBM, BelanjaQu, Binary Academy, Dropsuite,
  Forstok, GO-JEK, Jualo, Jurnal, Karir.com,
  KlikDokter, KMK Online, Mekari, Mitrais,
  Peentar, Qontak, Sejasa.com, Sleekr,
  Sribulancer, Starqle, Virkea, Wego, WGS ]
```

Source: :id_ruby - Komunitas Ruby Indonesia - https://ruby.id

> Is Ruby Dead Programming Language?

```
Language::Ruby.is_dead?(
  "Ruby is alive and thinking about next 25 years"
  "dev.to/teaglebuilt/is-ruby-dead-17em"
  "hackernoon.com/the-state-of-ruby-2019-is-it-dying"
  "GitHub Blog - Upgrading GitHub to Ruby 2.7"
  "..."
)
⇒ false
```

> Is Ruby Dead Programming Language?

```
Framework::Rails.is_dead?(
  "similartech.com/technologies/ruby-on-rails"
  "similartech.com/categories/framework"
  "hackernoon.com/is-ruby-on-rails-dead-in-2019"
  "dockyard.com/blog/ruby-on-rails-in-2020"
  "Is Ruby on Rails dying: Myth or Reality?"
  "..."
)
⇒ false
```

> Is Ruby Dead Programming Language?

```
Language::Ruby.is_dead?(
  "Ruby is dead. (rubyisdead.science)"
)
⇒ false!

@jmcharnes.said("isrubydead.com")
⇒ NO
```

Just because Ruby is not popular anymore, it doesn't mean Ruby is dead, right?

> Kenapa Memilih Ruby?

```
Learning::Ruby.is_worth?(:me).value
{
a: "Membantu menterjemahkan ide/solusi, menjadi sebuah
    script atau program."
b: "Membantu menyederhanakan kerumitan."
c: "Membantu menjembatani interaksi user dengan sistem
    Operasi."
d: "..."
} ⇒ Handy & Practical
```

> Kenapa Memilih Ruby?

```
PracticalThing::Ruby.can_do(:you)
⇒ {
a: "Full-stack web development"
b: "Web scraping & crawling"
c: "Static website generators"
d: "Automation, Backup, & DevOps tools"
e: "Build your own servers"
f: "Parsing data, cleaning & filtering"
g: "API clients (like Twitter's or GitHub's)"
h: "Report generators (PDF, HTML, CSV, SpreadSheet)"
i: "Command-line tools"
j: "Games (gosu, Ruby2D, etc.)"
h: "Data Science"
l: "Machine Learning & AI" }
```

Source: RubyGuide - Jesus Castello - https://www.rubyguides.com/2019/11/what-can-you-do-with-ruby/

> What's new on Ruby 2.7?

```
Ruby.whats_new?(2.7)
⇒ {
a: "Enumerable#tally"
b: "Numbered params for Blocks [Experimental]"
c: "Array#intersection"
d: "Enumerable#filter_map"
e: "Enumerator#produce"
f: "IRB gets a face lift"
g: "Ruby patern matching [Experimental]"
h: "10x performance ↑ for fiber & thread creattoin"
i: "Others changes, *go to source" }
```

Source: RubyGuide - Jesus Castello - https://www.rubyguides.com/2019/12/ruby-2-7-new-features/

> What’s new on Ruby 2.7?

Ruby NEVER STOPS Improving!

Now, according to Matz…

This is the last 2.x version release.
Because next year,
we’re getting Ruby 3.0!

Source: RubyGuide - Jesus Castello - https://www.rubyguides.com/2019/12/ruby-2-7-new-features/

> Tips Memasang Ruby for Dev

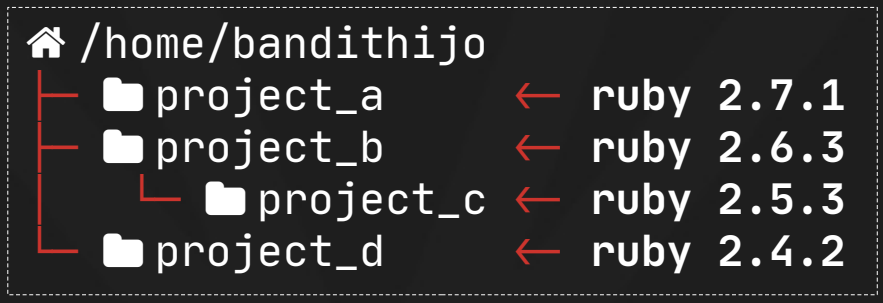
Groom your app's Ruby env. with Rbenv

Basic GitHub Checkout way:

```
$ cd
$ git clone https://github.com/rbenv/rbenv.git ~/.rbenv
$ echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc # ~/.zshrc
$ echo 'eval "$(rbenv init -)"' >> ~/.bashrc # ~/.zshrc
$ exec $SHELL

$ git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build
$ echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc
                                                                  # ~/.zshrc
$ exec $SHELL

$ rbenv install 2.7.1
$ rbenv global 2.7.1
$ ruby -v
```



Source: Rbenv (GitHub) - https://github.com/rbenv/rbenv/blob/master/README.md

Source: GoRails.com - https://gorails.com/setup/ubuntu/20.04#ruby-rbenv

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.IRB (Interactive Ruby Shell)

Ruby for Beginner
A Programmer's Best Friend

> Apa itu IRB?

IRB adalah kependekan dari Interactive Ruby.
Sebuah shell yang dapat kita gunakan untuk bermain-main dan bereksplorasi dengan Ruby dan langsung melihat hasilnya.
Definisi ini diistilahkan dengan REPL (Read-Eval-Print-Loop).

Untuk menjalankan IRB, cukup tulis di terminal dengan perintah:

```
$ irb
irb(main):001:0> _
```

Ayo kita coba jalanakan operasi matematika!

```
irb(main):001:0> 1 + 2 * ( 3**4 ) / 5 - 6
⇒ 2
```

Source: RubyGuides - https://www.rubyguides.com/2018/12/what-is-a-repl-in-ruby/

> Customisasi IRB

Secara default, bentuk dari “prompt” IRB akan seperti ini

```
$ irb
irb(main):001:0> _
```

Secara pribadi, saya kurang suka melihat prompt yang terlalu panjang.
Kita bisa menyederhanakan dengan memanggil dengan cara berbeda.

```
$ irb --prompt simple
>> _
```

Untuk membuat prompt “simple” menjadi default yang digunakan, cukup tambahkan pada file ~/.irbc (buka dengan text editor).

```
IRB.conf[:PROMPT_MODE] = :SIMPLE
```

Nah, sekarang apabila teman-teman memanggil irb, maka otomatis akan menggunakan “simple” prompt.

Source: Rake Routes - https://www.rakeroutes.com/customize-your-irb/

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Variabel

Ruby for Beginner
A Programmer's Best Friend

> Apa itu Variabel?

Digunakan untuk memberikan nama & menyimpan nilai (value)

In Ruby, Variable is **just a label**.
...a name for something that you can use to reference this value in your Ruby programs.

Just like the name we give to real-world things.
When I say "apple" you know what I'm talking about.
I don't have to describe it to you.
That's what variables do!

```
nama_lengkap = 'Yukihiro Matsumoto'    ← String
tahun_lahir  = 1995                    ← Integer
penjajahan   = 3.5                     ← Float
bumi_datar   = false                   ← Boolean
kamar_kosong = nil                     ← NilClass
tim_dev      = ['budi', 1, true]       ← Array
kontak_dev   = { 'nama': 'budi' }      ← Hash
```

Source: RubyGuides - https://www.rubyguides.com/ruby-tutorial/variables/

> Mengecek Class dari Variabel

Darimana kita tahu **variable** tertentu termasuk dalam Class apa?

Menggunakan .class

```
>> nama = 'Yukihiro Matsumoto'
>> nama.class
⇒ String

>> umur = 55
>> umur.class
⇒ Integer

>> tim_dev = ['Rizal', 'Reza', 'Rizqi']
>> tim_dev.class
⇒ Array
```

> Scope/jangkauan Variabel?

Local variable

```
team_name   = "Elite Lokal"
```

Global variable (prefix: $)

```
$team_name  = "Elite Global"
```

Instance variable (prefix: @)

```
@fav_food   = "Indomie Instance"
```

Class variable (prefix: @@)

```
@@fav_drama = "Itaewon Class"
```

Constant variable (Huruf Besar)

```
PHI = 3.14
```

Source: RubyGuides - https://www.rubyguides.com/ruby-tutorial/variables/

> Scope/jangkauan Variabel?

The difference between them?
It's on their "**scope**".

A variable scope answers this question:

> "***From where can I access this variable?***"

This is only going to matter when you start learning about Object-Oriented Programming.

Source: RubyGuides - https://www.rubyguides.com/ruby-tutorial/variables/
Source: RubyGuides - https://www.rubyguides.com/2019/03/ruby-scope-binding/

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Working with String (a)

Ruby for Beginner
A Programmer's Best Friend

.Working with String (a)

> String Concatenation
> String Interpolation
> Mengecek Method pada Class
> String Mehod yang Biasa Digunakan
> Escaping Character

> String Concatenation

Combining multiple string

a. Menggunakan operator +

```
>> nama_depan    = 'Yukihiro'
>> nama_belakang = 'Matsumoto'
>> nama_lengkap  = nama_depan + nama_belakang

>> puts nama_lengkap
"YukihiroMatsumoto"

>> puts nama_depan + ' ' + nama_belakang
"Yukihiro Matsumoto"

>> puts 'Yukihiro' + ' ' + 'Matsumoto'
"Yukihiro Matsumoto"
```

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/

> String Concatenation

Combining multiple string

b. Menggunakan operator +=

```
>> nama = ''

>> nama += 'Yukihiro'
>> nama += ' '
>> nama += 'Matsumoto'

>> puts nama
"Yukihiro Matsumoto"
```

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/

> String Concatenation

Combining multiple string

c. Menggunakan Ruby concat method

```
>> nama = ''

>> nama.concat('Yukihiro')
>> nama.concat(' ')
>> nama.concat('Matsumoto')

>> puts nama
"Yukihiro Matsumoto"
```

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/

> String Concatenation

Combining multiple string

d. Menggunakan operator << (alias dari Ruby concat method)

```
>> nama = ''

>> nama << 'Yukihiro'
>> nama << ' '
>> nama << 'Matsumoto'

>> puts nama
"Yukihiro Matsumoto"

>> puts nama << 'Yukihiro' << ' ' << 'Matsumoto'
"Yukihiro Matsumoto"
```

```
>> 'nomor rumah = ' << 1
# TypeError: no implicit conversion of Fixnum into String
```

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/

> String Concatenation

Combining multiple string

d. Menggunakan operator << (alias dari Ruby concat method)

Problem:

```
>> 'nomor rumah = ' << 1
=> "nomor rumah = \u0001"
```

Solusi:

```
>> 'nomor rumah = ' << 1.to_s
=> "nomor rumah = 1"
```

Menggunakan **to_s** method untuk mengkonversi Integer ke String.

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/
Source: RubyGuides - https://www.rubyguides.com/2018/09/ruby-conversion-methods/

> String Concatenation

Combining multiple string

e. Menggunakan Ruby prepend method (kebalikan concat)

```
>> nama = ''

>> nama.prepend('Yukihiro')
>> nama.prepend(' ')
>> nama.prepend('Matsumoto')

>> puts nama
"Matsumoto Yukihiro"
```

If you're thinking there is an **append** method,
well there isn't for String.
append method only for arrays.

Source: RubyGuides - https://www.rubyguides.com/2019/07/ruby-string-concatenation/

> String Interpolation

Interpolation or merging of variables into strings is a powerful technique. It allows you to "templatize" a string.

a. Menggunakan “#{}”

```
>> nama_depan    = ‘Yukihiro’
>> nama_belakang = ‘Matsumoto’
>> umur_sekarang = 55

>> puts “Nama: #{nama_depan} #{nama_belakang}. Umur: #{umur}”
“Nama: Yukihiro Matsumoto. Umur: 55”
```

Syarat: String harus menggunakan tanda petik ganda (“...”)

Source: RubyGuides - https://www.rubyguides.com/ruby-tutorial/variables/

> Mengecek Method pada Class

Darimana kita tahu **method** apa yang tersedia dari sebuah Class?

Menggunakan .methods

```
>> nama = 'Yukihiro Matsumoto'
>> nama.class
⇒ String

>> nama.methods
⇒[ :unicode_normalize, :unicode_normalize!, :ascii_only?, :unpack,
:unpack1, :to_r, :shellsplit, :encode, :encode!, :shellescape, :
%, :include?, :*, :+, :pretty_print, ..., ..., ... ]
```

Berapa banyak jumlah method yang tersedia? (misal: String)

```
>> nama.class
String

>> nama.methods.count
⇒ 192
```

> String Method yang Biasa Digunakan

.upcase()

```
>> nama = 'Yukihiro Matsumoto'
>> nama.upcase
⇒ "YUKIHIRO MATSUMOTO"
```

.downcase()

```
>> nama = 'Yukihiro Matsumoto'
>> nama.downcase
⇒ "yukihiro matsumoto"
```

.capitalize()

```
>> nama = 'yukihiro matsumoto'
>> nama.capitalize
⇒ "Yukihiro matsumoto"
```

.reverse()

```
>> nama = 'yukihiro matsumoto'
>> nama.reverse
⇒ "otomustam orihikuy"
```

.length()

```
>> nama = 'Yukihiro Matsumoto'
>> nama.length
⇒ 18
```

More: Ruby-Doc - https://ruby-doc.org/core-2.7.0/String.html

> String Method yang Biasa Digunakan

.sub()

```
>> nama = 'Yukihiro Matz'
>> nama.sub('Matz', 'Akita')
⇒ "Yukihiro Akita"
```

.gsub()

```
>> nama = 'Budi Budi Bayu'
>> nama.gsub('Budi', 'Bayu')
⇒ "Bayu Bayu Bayu"
```

.strip()

```
>> nama = '    matsumoto    '
>> nama.strip
⇒ "matsumoto"
```

.split()

```
>> nama = 'Budi Budi Bayu'
>> nama.split(' ')
⇒ ["Budi", "Budi", "Bayu"]
```

dan masih banyak lagi...

More: Ruby-Doc - https://ruby-doc.org/core-2.7.0/String.html

> Escaping Character

Bagaimana cara melakukan escaping character pada String?

```
>> 'malam jum'at'
SyntaxError: unexpected local variable or method...

>> 'malam jum\'at'
"malam jum'at"
```

Alternatif gunakan **double** quote untuk escaping **single** quote:

```
>> "malam jum'at"
"malam jum'at"
```

Contoh-contoh escaping character

```
\' : single quote
\" : double quote
\\ : backslash
\n : new line
\r : carriage return
```

```
\t : tab
\b : backspace
\f : form feed
\v : vertical tab
est.
```

More: AppSignal - https://blog.appsignal.com/2016/12/21/ruby-magic-escaping-in-ruby.html

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Working with String (b)

Ruby for Beginner
A Programmer's Best Friend

.Working with String (b)

> Get Input from User (Console)

> Get Input from User (Console)

Ruby use **gets** method to get the user input (as a string).

.gets

```
>> puts “Nama kamu siapa? “
>> nama = gets
Yukihiro
⇒ “Yukihiro\n”
```

“\n” didapatkan saat kita menekan tombol enter.
Bagaimana cara mengilangkannya?

Gunakan **String#chomp** method.

.gets.chomp

```
>> puts “Nama kamu siapa? “
>> nama = gets.chomp
Yukihiro
⇒ “Yukihiro”
```

Source: Ruby-Doc - https://ruby-doc.org/core-2.7.1/IO.html

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Working with Number

Ruby for Beginner
A Programmer's Best Friend

.Working with Number

> Arithmetic Operation
> Order Arithmetic Operation
> Loop with Times
> Upto & Downto

> Arithmetic Operation

Nothing new here…

```
>> 5 + 5
=> 10

>> 10 * 2
=> 20

>> 2 ** 3
=> 8

>> 10 % 2
=> 0

>> 10.even?
=> true

>> 4.odd?
=> false
```

```
>> 10 / 2.0
=> 5.0

>> 9.0 / 3
=> 3.0
```

Source: Ruby-Doc - https://ruby-doc.org/core-2.7.1/IO.html

> Order Arithmetic Operation

Perhatikan urutan dari pengerjaan operator.

Gunakan jembatan “**PEMDAS**” untuk mengingatnya.

Parenthesis	()
Exponential	**
Multiplication	*
Division	/
Addition	+
Substraction	-

Source: Ruby-Doc - https://ruby-doc.org/core-2.7.1/IO.html

> Loop with times

Salah satu method yang mudah untuk melakukan looping / perulangan terhadap angka.

.times

```
?> 3.times do
?>   puts 'HelloDev'
>> end

HelloDev
HelloDev
HelloDev
```

.times (one line)

```
>> 3.times { puts 'HelloDev' }

HelloDev
HelloDev
HelloDev
```

```
?> 2.times do |n|
?>   puts n
>> end

0
1
```

```
>> 2.times do { |n| puts n }

0
1
```

> Upto & Downto

Salah satu method yang mudah untuk melakukan looping / perulangan terhadap angka dengan rentang tertentu.

.upto().each

```
?> 1.upto(3).each do
?>   puts 'HelloDev'
>> end

HelloDev
HelloDev
HelloDev
```

.downto().each

```
?> 3.downto(1).each do
?>   puts 'HelloDev'
>> end

HelloDev
HelloDev
HelloDev
```

```
?> 1.upto(2) do |n|
?>   puts n
>> end

1
2
```

```
?> 2.downto(1).each do |n|
?>   puts n
>> end

1
2
```

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Comparison Operators

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Branching /Condition

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Collection

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Looping

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Method

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

EXTENDED

Ruby for Beginner

A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.File IO

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Error Handling

Ruby for Beginner
A Programmer's Best Friend

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

.Object Oriented Programming

Ruby for Beginner
A Programmer's Best Friend

hello dev

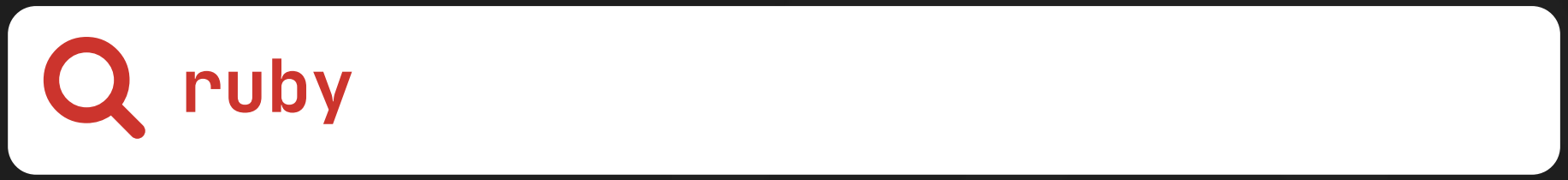
#ONLINEMEETING
2020-09-11 20.00 WITA

.Module

Ruby for Beginner
A Programmer's Best Friend

> Tips mencari dengan Google

Gunakan prefix “ruby” sebagai awalan kata kunci untu mencari di Google.



Kemudian, diikuti dengan keyword apa yang mau dicari.
Misal, **tentang perulangan dengan method “each”.**

Dengan cara ini, kita dapat membuat hasil pencarian yang lebih spesifik untuk bahasa pemrograman Ruby.

> Referensi Belajar Ruby

YouTube:
- GoRails
- Jesus Castello
- Drifting Ruby
- Nate Berkopec
- TechmakerTV
- Decypher Media
- zayne
- Code School
- FreeCodeCamp.org
- Sekolah Koding
- Agung Setiawan

WebPage:
- Ruby in 20 minutes
- RubyGuides.com

Online Course:
- IDRails.com

> Referensi Group Telegram untuk Bertanya seputar Ruby

- Ruby Indonesia
- Rails Indonesia

> Referensi Channel Telegram untuk Berita tentang Ruby

- Ruby/Rails Inside

hello dev

#ONLINEMEETING
2020-09-11 20.00 WITA

:Terima_Kasih

> Made with ♥ by Rizqi Nur Assyaufi
@BanditHijo
2020/09

Ruby for Beginner
A Programmer's Best Friend