

---

# **Panduan Umum Praxis Academy**

Untuk siswa Praxis Academy

Dr. Bambang Purnomosidi D. P.



# Daftar Isi

<b>1</b>	<b>Tentang Buku Ini</b>	<b>3</b>
<b>2</b>	<b>Proses Pendidikan</b>	<b>4</b>
	Pendahuluan . . . . .	4
	Ketentuan Umum . . . . .	4
	Tahap Pendidikan . . . . .	4
	Tahap Incubator . . . . .	5
	Novice . . . . .	5
	Incubator - Initiator . . . . .	8
	Incubator - Team Player . . . . .	8
<b>3</b>	<b>Kurikulum</b>	<b>10</b>
	Full Stack Enterprise Application Development . . . . .	10
	Minggu 01 . . . . .	10
	Minggu 02 . . . . .	11
	Minggu 03 . . . . .	11
	Minggu 04 . . . . .	12
<b>4</b>	<b>Resources</b>	<b>13</b>

# 1 Tentang Buku Ini



## Versi

1 1.0.0-RC--28-Mei-2019--21:58:44

Buku ini merupakan buku pegangan para mentor, siswa, serta pihak-pihak lain yang berkepentingan dengan proses pendidikan di Praxis Academy. Dengan panduan buku ini, diharapkan proses belajar menjadi lebih efektif dan efisien serta mempunyai pedoman yang jelas. Lisensi buku ini adalah Creative Commons Attributiona ShareAlike 4.0 International License - CC-BY-SA 4.0. Secara umum, penggunaan lisensi ini mempunyai implikasi bahwa pengguna materi:

1. Harus memberikan atribusi ke penulis dan sponsor untuk penulisan materi ini (Praxis Academy).
2. Boleh menggunakan produk yang ada disini untuk keperluan apapun jika point 1 di atas terpenuhi.
3. Boleh membuat produk derivatif dari produk yang ada disini sepanjang perubahan-perubahan yang dilakukan diberitahukan ke kami dan di-share dengan menggunakan lisensi yang sama.

Untuk penggunaan selain ketentuan tersebut, silahkan menghubungi:

1 Praxis Academy  
2 Jl.Garuda no 67, Manukan  
3 Condong Catur  
4 Sleman  
5 Yogyakarta 55283  
6 Indonesia  
7 Email: [hello@praxisacademy.id](mailto:hello@praxisacademy.id)  
8 Web: <https://praxisacademy.id>

## 2 Proses Pendidikan

### Pendahuluan

Dunia software dan pemrograman merupakan dunia yang luas. Tidak memungkinkan untuk mengajarkan dan memberikan bekal ke siswa tentang semua hal. Oleh karena itu, sekolah ini lebih menekankan pada kekuatan konsep sehingga memudahkan adaptasi siswa ke dunia industri. Konsep yang kuat tidak dilakukan dengan cara memberikan kuliah-kuliah seperti pada perguruan tinggi tetapi melalui latihan dan pembelajaran praktik yang intensif dengan pemahaman ke arah konsep.

### Ketentuan Umum

- Pendidikan dilaksanakan pada hari kerja (senin - jumat), dari jam 09:00 - 17:00.
- Peserta harus membawa laptop sendiri, disarankan memory minimal 8 GB, OS yang didukung hanya Linux (terutama Ubuntu), siswa boleh menggunakan OS apapun, tetapi jika tidak sesuai dengan yang didukung (Ubuntu), tugas siswa sendiri untuk mengelola dan mencari resources serta solusi sendiri terkait OS tersebut (misalnya untuk instalasi software, dan lain-lain).
- Kehadiran sifatnya mutlak. Ketidakhadiran akan menjadi perhatian khusus dan bisa mengarah ke dihentikannya peserta dari proses pendidikan. Masalah seperti ini akan menjadi perhatian internal dan akan dirapatkan oleh internal Praxis Academy untuk tindak lanjut dari kondisi tersebut.
- Waktu untuk beribadah tidak boleh diganggu. Siswa dipersilahkan meninggalkan kelas untuk mengerjakan ibadah. Jam untuk ibadah ini tidak ditulis secara khusus tetapi siswa dipersilahkan melaksanakan sesuai dengan jam masing-masing.

### Tahap Pendidikan

Tahap pendidikan ada 3:

- Tahap *novice*: siswa dianggap mempunyai cukup dasar untuk masuk ke proses. Tahap ini berisi

pemberian teori, praktik, dan latihan intensif (1 bulan pertama) dengan tujuan untuk membentuk mental dan skill *developer*.

- Tahap *incubator*: siswa dianggap siap untuk masuk ke industri. Tahap ini mempunyai 2 kemungkinan, masing-masing akan dijelaskan berikut ini.

## Tahap Incubator

Pada tahap ini, siswa akan mendapatkan tantangan mengerjakan software secara langsung. Ada 2 kemungkinan untuk tahap ini:

- **Bergabung ke tim pengembang software dalam bentuk *internship*.** Kemungkinan ini terjadi jika ada proyek riil pengembangan software. Siswa dilibatkan secara langsung dalam tim dan dibimbing untuk mengerjakan proyek riil tersebut.
- **Proyek mandiri.** Kemungkinan ini terjadi jika tidak ada proyek riil pengembangan software. Situasi ini pada dasarnya mempunyai 2 sub tahap, yaitu tahap *initiator* (siswa dianggap mempunyai kemampuan teknis yang mencukupi dan siap untuk mempunyai inisiatif dalam membuat software tertentu, siswa berinisiatif membuat software sesuai keinginan pribadi) dan tahap *team player* (siswa dianggap mempunyai kemampuan individual yang memadai dan bisa menggunakan kemampuan individual mereka untuk pembuatan software secara berkelompok). kedua sub tahap ini dilaksanakan dalam waktu masing-masing 2 minggu.

Siswa diminta semaksimal mungkin berkontribusi ke proyek-proyek open source yang ada di GitHub.

## Novice

### Tujuan

Memberikan dasar teori dan praktik yang kuat terhadap materi

### Aktivitas Harian

Setiap hari, akan dilakukan pemberian materi serta latihan intensif dengan alokasi waktu sebagai berikut:

- Sebelum jam 9: presensi kedatangan
- Jam 9 - 10: pemberian materi, penjelasan serta contoh-contoh oleh mentor.
- Jam 10 - 12: latihan materi dari mentor.
- Jam 13 - 17: latihan kasus lebih kompleks.
- jam 17: presensi pulang.

Setelah selesai, hasil untuk setiap hari di-push ke repo GitHub dengan nama repo **praxis-academy**. Pada repo tersebut, buat direktori *novice* dan *incubator*. Pada direktori *novice*, buat direktori sesuai dengan minggu dan hari pelaksanaan:

## 1 Minggu-Hari

- Minggu: Minggu saat pengerjaan (2 digit)
- Hari: Hari saat pengerjaan (2 digit)

Contoh:

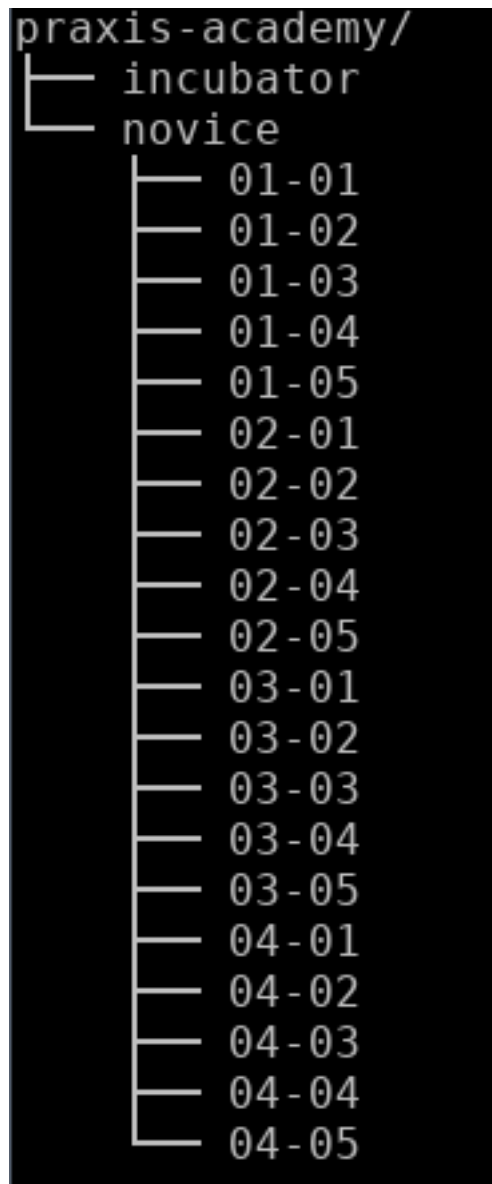
02-03: repo untuk minggu ke 2, hari ke 3 (*rabu*).

Pada repo tersebut, buat `README.md` dengan isi minimal:

- Judul: tentang materi hari tersebut
- Oleh: nama siswa
- Tanggal
- Ringkasan materi
- Penjelasan tentang isi repo
- Lisensi

Pada repo tersebut, isikan juga berbagai hasil file yang dikerjakan pada hari itu dengan struktur bebas. File serta strukturnya dijelaskan di dalam file `README.md` di atas.

Berikut adalah contoh direktori yang dibuat:



**Gambar 2.1:** Struktur direktori

Peserta diwajibkan posting ke salah satu akun media sosial peserta minimal 1x dalam seminggu, berisi materi yang sudah ditulis di dalam GitHub tersebut dan menyertakan URL ke repo yang mereka buat. Pada akhir bulan, 1 hari terakhir di tahap ini digunakan untuk evaluasi metrik penilaian. Ada proses untuk menguji pemahaman dan penguasaan siswa terhadap materi:

- Semua materi GitHub

Mentor melakukan proses pengujian menggunakan mekanisme:

- Ujian teori (tes lisan terkait pemahaman siswa).

- Ujian praktik (tes coding / programming).
- Hasil dari pengujian ini adalah kelanjutan siswa di tahap berikutnya: melanjutkan atau gagal.

Pengujian ini dilaksanakan pada hari Sabtu.

Materi yang dibahas ada pada bab berikutnya.

## **Incubator - Initiator**

### *Tujuan*

Siswa membuat software. Hasil akhir dari proses ini adalah software open source yang diletakkan pada repo Git (GitHub) serta deployment dari software tersebut (di Google PlayStore untuk aplikasi Android, di DockerHub untuk image Docker, binary release di Github Releases, dan lain-lain). Software harus direncanakan dengan baik dalam bentuk milestones dan tiket-tiket untuk milestones (dari awal sampai rilis). File README.md diletakkan pada repo untuk menjelaskan tentang isi dari repo. Arsitektur software, milestones, dan dokumentasi tentang software tersebut diletakkan di berbagai file berformat Markdown di direktori docs.

### *Minggu pertama*

- Hari 1 dan 2: siswa mencari tema dari proyek software yang akan dibuat.
- Hari 3: siswa mendiskusikan dengan mentor terkait software yang akan dibuat, fitur-fitur, serta milestones untuk menyelesaikan software tersebut. Siswa (dengan bantuan mentor) menyiapkan infrastruktur untuk pembuatan software.
- Akhir hari 3: dibuat repo dan dalam repo tersebut dibuat README terkait no 2 di atas.
- Hari 4: siswa mulai mengerjakan pembuatan software.

### *Minggu 2*

Mengerjakan pembuatan software dan melengkapi dokumentasi. GitHub releases berisi berbagai rilis (versi Beta, RC1, RC2, dan seterusnya sampai dengan rilis).

### *Hari Terakhir Minggu 2*

Siswa mempresentasikan pekerjaan pembuatan software ini. Tim akademik dan mentor melakukan penilaian serta pengujian terhadap software yang dibuat tersebut. Hasil penilaian ini menentukan apakah siswa berhasil masuk ke tahap berikutnya atau gagal.

## **Incubator - Team Player**

### *Tujuan*



Tahap ini digunakan untuk melatih kemampuan bekerjasama antar siswa. Team ditentukan oleh akademik dan terdiri atas 3-5 siswa. Tim telah dibentuk sebelum memasuki tahap ini. Pada tahap ini, tim membuat software untuk dinilai.

#### *Minggu 1*

- Hari 1 dan 2: tim melakukan brainstorming untuk menentukan software apa yang akan dibuat.
- Hari 2: pemberian materi tentang software development methodology serta penggunaan GitHub untuk mengelola proyek bersama.
- Hari 3: tim mendiskusikan dengan mentor terkait software yang akan dibuat, fitur-fitur, serta milestones untuk menyelesaikan software tersebut. tim (dengan bantuan mentor) menyiapkan infrastruktur untuk pembuatan software.
- Akhir hari 3: dibuat repo dan dalam repo tersebut dibuat README terkait no 2 di atas.
- Hari 4: siswa mulai mengerjakan pembuatan software. Hari 3:

#### *Hari-hari berikutnya sampai akhir minggu ke 3 dan minggu 4*

- Iterasi pengerjaan proyek
- Siswa mengikuti ritme pengerjaan proyek, termasuk mengalami deadline, pull request ditolak (jika ada), stand up meeting, dan lain-lain.

#### *Akhir Minggu 4*

- Hari Jumat, bagian akademik serta mentor melakukan evaluasi komprehensif terhadap kemampuan dari siswa dengan melihat hasil GitHub personal maupun tim. Pada hari ini, tim mempresentasikan hasil kerja kelompok.
- Siswa menjalani proses ujian terkait tahap newbie, singledev, dan teamwork.
- Bagian akademik serta para mentor memutuskan siswa yang berhasil lulus dari Praxis Academy serta siswa yang tidak berhasil lulus dari Praxis Academy.

## 3 Kurikulum

Materi yang terdapat pada bab ini merupakan materi yang akan dikerjakan pada tahap **Novice** (lihat bab sebelumnya). Rincian target pembelajaran ada pada repo dan buku panduan tersendiri.

### Full Stack Enterprise Application Development

#### Minggu 01

##### *Hari 1*

1. Teknologi Software
2. Sistem Operasi: UI, shell, utilities

##### *Hari 2*

1. Git dan GitHub
2. Git untuk single user
3. Git untuk *team*

##### *Hari 3*

1. Development tools dan ekosistemnya
2. Dasar-dasar Java

##### *Hari 4*

1. Package di Java<
2. Dynamic typing dan static typing
3. Variabel, konstanta, operator, ekspresi, statement, dan tipe data di Java
4. Penanganan error / exception

##### *Hari 5*

1. Dasar-dasar Pemrograman Berorientasi Obyek di Java<
2. Alur kendali program di Java
3. Algoritma, pseudocode, dan implementasi algoritma ke Java

**Minggu 02***Hari 1*

1. Annotations
2. Object Oriented Programming Lanjut di Java

*Hari 2*

1. Generics
2. Functional Programming di Java

*Hari 3*

1. Unit Testing di Java

*Hari 4*

1. Serialisasi data: XML, JSON, dll
2. Struktur data di Java

*Hari 5*

1. Concurrent Programming di Java

**Minggu 03***Hari 1*

1. Pustaka standar dan pustaka pihak ketiga di Java
2. Proses *build* dan pengelolaan paket pustaka di Java: Gradle

*Hari 2*

1. Database management system
2. Akses ke database menggunakan Java: JDBC dan ORM

*Hari 3*

1. NoSQL dan NewSQL

*Hari 4*

1. RESTful endpoint
2. GraphQL

*Hari 5*

1. Arsitektur Software: microservices dan *full stack applicatio*
2. Framework: Dasar-dasar Micronaut

## **Minggu 04**

### *Hari 1*

1. Berpindah development tools: step-by-step
2. Studi kasus perpindahan development tools: dari Java ke peranti pengembangan lainnya (JavaScript, Kotlin, Go, PHP)
3. Dasar-dasar JavaScript dan Node.js

### *Hari 2*

1. JavaScript

### *Hari 3*

1. Vue.js

### *Hari 4*

1. Aplikasi Full stack: Micronaut + Vue.js
2. Aplikasi Full stack: Micronaut + Vue.js + Database

## 4 Resources

Menjadi *software engineer / software developer / developer / programmer / coder* bukan hal yang bisa dipelajari hanya dalam semalam. Bagian ini memberikan beberapa *resources* yang baik bagi mereka yang (ingin) berkarir di bidang *software development / engineering*.

1. Professional Programming, repo GitHub yang berisi banyak sekali *resources* untuk membuat seseorang menjadi *software developer* yang kompeten.
2. Awesome Awesomeness, repo GitHub yang berisi petunjuk ke koleksi-koleksi *resources* yang terkait dengan peranti pengembangan dalam berbagai bahasa pemrograman serta hal-hal terkait lainnya.