# Kemampuan Dasar

## Memahami Ruang Lingkup Pengembangan Software

1. Teknologi informasi dan teknologi software
2. Metodologi Pengembangan Software

### Tujuan

1. Memahami teknologi informasi secara umum dan kaitannya dengan software.
2. Memahami ranah pendidikan yang terkait dengan teknologi informasi: istilah teknik informatika / informatics, ilmu komputer / computer science.
3. Memahami arti penting dari metodologi pengembangan software
4. Memahami Agile Software Development secara umum
5. Memahami dasar yang kuat dari praktik Agile Software Development: Extreme Programming dan Scrum

### Pembahasan

1. Teknologi informasi dan teknologi software
2. Metodologi pengembangan software
3. Extreme Programming dan Scrum

### Pembelajaran

Materi dan Penjelasan

1. Halaman [Wikipedia - Information Technology](https://en.wikipedia.org/wiki/Information_technology), serta komponen-komponen dari TI dengan penekanan lebih pada [software](https://en.wikipedia.org/wiki/Software).
2. Halaman [StackExchange tentang perbedaan informatics dengan computer science](https://cs.stackexchange.com/questions/81408/whats-the-difference-between-computer-science-and-informatics).
3. Halaman [Wikipedia - Software Development](https://en.wikipedia.org/wiki/Software_development) memberikan penjelasan umum tentang aktivitas, model, serta hal-hal lain yang terkait dengan software development. Dengan mempelajari bagian ini, siswa diharapkan memahami bahwa software development merupakan aktivitas yang kompleks dan memerlukan berbagai macam roles / peran. Penerapan dari engineering ke dalam aktivitas software development ini disebut dengan [Software Engineering](https://en.wikipedia.org/wiki/Software_engineering)
4. Halaman [Wikipedia - Agile Software Development](https://en.wikipedia.org/wiki/Agile_software_development) membahas hal yang lebih spesifik dari software development, yaitu salah satu model / metodologi yang digunakan untuk membangun software.
5. Halaman [Wikipedia - Scrum](https://en.wikipedia.org/wiki/Scrum_(software_development)) membahas tentang salah satu process framework yang termasuk dalam metodologi Agile.
6. Halaman [Wikipedia - XP](https://en.wikipedia.org/wiki/Extreme_programming) membahas tentang salah satu process framework yang termasuk dalam metodologi Agile.

## Kemampuan Teknis Dasar 1: Sistem Operasi

1. Sistem Operasi: UI, shell, utilities, shell script

### Tujuan

1. Memahami software paling mendasar: operating system
2. Memahami ekosistem sistem operasi saat ini
3. Memahami komponen-komponen sistem operasi - khususnya berbasis Linux
4. Memahami UI di Linux (GUI dan text mode / shell)
5. Memahami dan bisa menggunakan berbagai utilitas dasar dalam Linux
6. Mehamami dasar-dasar dari shell script.

### Pembahasan

1. Cara software bekerja
2. Ekosistem sistem operasi saat ini
3. Komponen sistem operasi
4. Untuk pemakai Linux:
   * Gambaran umum Linux
   * Komponen-komponen SO Linux
   * Windowing System di Linux
   * Linux console dan terminal
   * Utilitas dasar Linux
   * Shell Script
5. Untuk pemakai macOS:
   * Gambaran umum macOS
   * Arsitektur Komponen-komponen macOS
6. Untuk pemakai Windows:
   * Keluarga Windows OS
   * Komponen-komponen Windows

### Pembelajaran

1. Cara [software](https://en.wikipedia.org/wiki/Software) menempati komputer kita serta bagaimana user berinteraksi dengan tasks komputasi. Tekankan pemahaman pada bagaimana [sistem operasi](https://en.wikipedia.org/wiki/Operating_system) bekerja.
2. Overview dari [daftar sistem operasi](https://en.wikipedia.org/wiki/List_of_operating_systems) yang ada di dunia ini.
3. Untuk pemakai Linux:
   * [Komponen-komponen SO Linux](https://en.wikipedia.org/wiki/Linux).
   * [Windowing System](https://en.wikipedia.org/wiki/Windowing_system) yang berfungsi sebagai antarmuka grafis (GUI) di Linux serta impelementasinya di Linux menggunakan [X Window System](https://en.wikipedia.org/wiki/X_Window_System).
   * Keterkaitan X Window System dengan [Display Manager](https://en.wikipedia.org/wiki/X_display_manager_(program_type)), [Window Manager](https://en.wikipedia.org/wiki/Window_manager), serta [Desktop Environment](https://en.wikipedia.org/wiki/Desktop_environment).
   * [Linux console](https://en.wikipedia.org/wiki/Linux_console) dan keterkaitannya dengan [Linux shell](https://en.wikipedia.org/wiki/Shell_(computing)).
   * Ringkasan dari [berbagai command shell](https://en.wikipedia.org/wiki/Comparison_of_command_shells) terutama yang digunakan di Linux.
   * Beberapa utilitas yang biasanya digunakan pada saat berada di console / shell. Beberapa utilitas tersebut biasanya berada dalam kelompok [GNU Core Utilities](https://www.gnu.org/software/coreutils/) dan [util-linux](https://en.wikipedia.org/wiki/Util-linux). Beberapa petunjuk lainnya adalah [Guru99](https://www.guru99.com/must-know-linux-commands.html) dan [E-Guide - format PDF](https://cse.yeditepe.edu.tr/~ayildiz/attachments/linux_commands_eguide.pdf).
   * Cara membuat [shell script](https://en.wikipedia.org/wiki/Shell_script).
4. Untuk pemakai macOS:
   * [Gambaran umum macOS](https://en.wikipedia.org/wiki/MacOS).
   * [Arsitektur macOS](https://en.wikipedia.org/wiki/Architecture_of_macOS).
   * [Terminal dan shell di macOS](https://www.techrepublic.com/blog/apple-in-the-enterprise/mac-terminal-basics-unlock-the-power-of-the-command-line/).
   * [Terminal di macOS](https://en.wikipedia.org/wiki/Terminal_(macOS)).
   * Dua shell yang sering dipakai di macOS: [Bash](https://www.gnu.org/software/bash/) dan [Zsh]((<http://zsh.sourceforge.net/>). Default shell di macOS adalah Bash (pre-Catalina). Mulai Catalina, shell default adalah Zsh.
   * [Beberapa utilities di macOS](https://www.mitchchn.me/2014/os-x-terminal/), lihat juga di [artikel ini](https://www.techrepublic.com/article/16-terminal-commands-every-user-should-know/).
   * [Daftar lengkap command line berbasis Bash di macOS](https://ss64.com/osx/).
   * [Shell script dan cara pembuatannya](https://en.wikipedia.org/wiki/Shell_script).
5. Untuk pemakai keluarga Microsoft Windows:
   * [Gambaran umum Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows).
   * [Gambaran umum Microsoft Windows 10](https://en.wikipedia.org/wiki/Windows_10).
   * [Gambaran umum arsitektur internal Windows](https://techcommunity.microsoft.com/t5/Ask-The-Performance-Team/Windows-Architecture-The-Basics/ba-p/372345).
   * [Windows Internal book](https://docs.microsoft.com/en-us/sysinternals/learn/windows-internals).
   * [Dokumentasi Powershell](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/powershell).
   * [Daftar command line di Windows](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands).
   * [Batch file](https://en.wikipedia.org/wiki/Batch_file).
   * [Powershell script](https://www.pdq.com/blog/writing-your-first-powershell-script/), lihat juga di [artikel ini](https://www.howtogeek.com/141495/geek-school-writing-your-first-full-powershell-script/) dan [artikel ini](https://www.guru99.com/powershell-tutorial.html).
   * [Daftar perintah Powershell](https://www.pdq.com/powershell/), lihat juga di [artikel ini](https://devblogs.microsoft.com/scripting/table-of-basic-powershell-commands/).
   * [Perintah-perintah di *batch file*](http://www.trytoprogram.com/batch-file-commands/).
   * [Batch file dan Powershell script](https://www.howtogeek.com/204088/how-to-use-a-batch-file-to-make-powershell-scripts-easier-to-run/).

Latihan

Catatan: semua hasil pada latihan ini disimpan pada direktori praxis-academy/kemampuan-dasar-1/latihan (jika menggunakan Windows, pemisah antar direktori adalah \, jadi pada Windows hasilnya akan disimpan di praxis-academy\kemampuan-dasar-1\latihan. Anda bebas menentukan lokasi tempat direktori praxis-academy berada.

Untuk Pemakai Linux

1. Boot laptop masing-masing, perhatikan proses booting dari awal, jika muncul Grub, usahakan melihat parameter dari Grub tersebut. Cari informasi di Internet tentang parameter Grub tersebut. Simpan hasil anda pada file teks biasa dengan nama file grub.txt.
2. Cari informasi tentang software yang ada di laptop anda: display manager yang digunakan, window manager yang digunakan, desktop environment yang digunakan, serta shell apa yang digunakan. Dari mana bisa mengetahui informasi tersebut? Tulis jawaban pada file teks biasa dengan nama file env.txt.
3. Cari lokasi dari kernel Linux, sebutkan file-file yang terkait dan kegunaannya. Tulis jawaban pada file teks biasa dengan nama file kernel-linux.txt.
4. Masuk ke terminal / console dan kerjakan beberapa perintah berikut melalui command line / shell:
   * silahkan coba beberapa perintah di [Guru99](https://www.guru99.com/must-know-linux-commands.html) dan [E-Guide - PDF](https://cse.yeditepe.edu.tr/~ayildiz/attachments/linux_commands_eguide.pdf), minimal 10 perintah.
   * tulis hasil dari masing-masing perintah tersebut ke dalam file cmdline.txt (gunakan copy paste dari shell)
5. Untuk latihan ini, letakkan hasil pada direktori praxis-academy/kemampuan-dasar-1/latihan/shellscripts: silahkan coba [30 contoh shell script](https://linuxhint.com/30_bash_script_examples/).

Untuk Pemakai macOS

1. Kerjakan berbagai hal yang ada di artikel tentang [beberapa utilities di macOS](https://www.mitchchn.me/2014/os-x-terminal/) dan [artikel ini](https://www.techrepublic.com/article/16-terminal-commands-every-user-should-know/). Tulis hasil semua yang anda coba pada file cmdline.txt.
2. Untuk latihan ini, letakkan hasil pada direktori praxis-academy/kemampuan-dasar-1/latihan/shellscripts: silahkan coba [30 contoh shell script](https://linuxhint.com/30_bash_script_examples/). Jika anda menggunakan Zsh, silahkan [kerjakan materi ini](https://www.linuxtoday.com/blog/writing-scripts-for-zsh.html).

Untuk Pemakai Windows

1. Kerjakan minimal 20 command line yang ada di [daftar command line di Windows](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands). Tulis hasil semua yang anda coba pada file cmdline.txt.
2. Untuk latihan ini, letakkan hasil pada direktori praxis-academy\kemampuan-dasar-1\latihan\powershellscripts. Kerjakan contoh [Powershell script](https://www.pdq.com/blog/writing-your-first-powershell-script/), juga di [artikel ini](https://www.howtogeek.com/141495/geek-school-writing-your-first-full-powershell-script/), [artikel ini](https://www.guru99.com/powershell-tutorial.html), dan [batch file dan Powershell script](https://www.howtogeek.com/204088/how-to-use-a-batch-file-to-make-powershell-scripts-easier-to-run/).

Kasus

Referensi:

1. [Wikibooks - Bash Shell Scripting](https://en.wikibooks.org/wiki/Bash_Shell_Scripting)
2. [Bash Reference Manual](https://www.gnu.org/software/bash/manual/bash.html)
3. [Bash Handbook](https://github.com/denysdovhan/bash-handbook)
4. [Dokumentasi Zsh](http://zsh.sourceforge.net/Doc/Release/zsh_toc.html)
5. [Perintah-perintah di *batch file*](http://www.trytoprogram.com/batch-file-commands/).
6. [Dokumentasi Powershell](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/powershell).

Selesaikan kasus-kasus berikut.

Catatan: semua hasil pada latihan ini disimpan pada direktori praxis-academy/kemampuan-dasar-1/kasus (jika menggunakan Windows, pemisah antar direktori adalah , jadi pada Windows hasilnya akan disimpan di praxis-academy\kemampuan-dasar-1\kasus. Anda bebas menentukan lokasi tempat direktori praxis-academy berada.

1. Buat shell script / powershell sript / batch file cari-java.sh (untuk shell script - macOS dan Linux) atau cari-java.bat (untuk Windows). Tujuan dari script tersebut adalah untuk melihat daftar file pada suatu direktori (termasuk direktori anak-anaknya) dan jika terdapat file dengan ekstensi .java - tampilkan tulisan "Ada file Java pada direktori {nama direktori}". Hasil eksekusi (di Linux, lainnya menyesuaikan):

$ cari-java.sh $HOME/src

Ada file Java pada direktori /home/bpdp/src/hari-01

1. Untuk pemakai Linux dan macOS: Buat shell script process-killer.sh untuk menanyakan suatu nama program (misalnya firefox), setelah itu mencari PID dari program tersebut dan jika PID program tersebut ada, maka program tersebut akan dimatikan. Saran: gunakan perintah-perintah ps, grep, awk, dan kill.
2. Untuk pemakai Windows: Lanjutkan kasus nomor 1. Beri nama file yang anda kerjakan ini dengan nama file-ren.bat. Setelah menemukan file .java, akan muncul pertanyaan:
   * nama-file.java ada, diganti namanya (y/t)?
   * jika y, maka input nama baru kemudian diganti
   * jika t, maka lanjut ke proses berikutnya

## Kemampuan Teknis Dasar 2: Source Code Management - Git dan GitHub

Bagian ini dikerjakan oleh semua pemakai sistem operasi: Linux, macOS, dan Windows.

### Tujuan

1. Memahami aktivitas pada metodologi agile serta kaitannya dengan source code management
2. Memahami dan mampu menggunakan Git untuk mengelola repository.
3. Memahami dan mampu menggunakan Git untuk mengelola remote repository di GitHub.
4. Memahami dan mampu menggunakan Git dan GitHub, baik untuk single person maupun team development.

### Pembahasan

1. Software engineering, software configuration management, version control, dan distributed version control.
2. Git dan perintah-perintah dasarnya
3. Markdown sebagai format untuk dokumentasi
4. Git dan remote repository (GitHub, GitLab, Assembla, BitBucket)
5. Git untuk single person development
6. Git untuk tim pengembang aplikasi

### Pembelajaran

Materi dan Penjelasan

1. Ruang lingkup [software engineering](https://en.wikipedia.org/wiki/Software_engineering).
2. Keterkaitan software engineering dengan [SCM - Software Configuration Management](https://en.wikipedia.org/wiki/Software_configuration_management).
3. Keterkaitan SCM dengan [Version Control](https://en.wikipedia.org/wiki/Version_control) dan [Distributed Version Control](https://en.wikipedia.org/wiki/Distributed_version_control).
4. [Git](https://en.wikipedia.org/wiki/Git) dan keterkaitannya dengan Distributed Version Control
5. [Men-*setup* Git - Chapter 1 - Getting Started](https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control), pastikan juga bahwa [git telah terinstall](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git).
6. [Membuat account GitHub](https://git-scm.com/book/en/v2/GitHub-Account-Setup-and-Configuration) serta [membuat repo di GitHub](https://help.github.com/en/articles/creating-a-new-repository).
7. [README.md](https://www.quora.com/What-is-README-MD-file-and-why-do-I-need-it) serta [mampu menggunakan pemformatan file markdown untuk menuliskan dokumentasi](https://help.github.com/en/articles/basic-writing-and-formatting-syntax).
8. Penggunaan git status, git add, git commit, git push untuk menyimpan ke remote repo
9. Penggunaan [branching and merging](https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging).
10. [Pull request](https://help.github.com/en/articles/about-pull-requests) untuk repo di GitHub milik sendiri, [merging](https://help.github.com/en/articles/merging-a-pull-request), kemudian sinkronisasi ke lokal repo di komputer.
11. [Menggunakan Github untuk kolaborasi tim](https://dev.acquia.com/blog/getting-started-collaborative-development-git).

Persiapan

Buat repo di account Github anda dengan nama praxis-academy. Setelah itu inisialisasi repo tersebut di komputer lokal anda. Repo ini yang akan menjadi hasil dari kemampuan dasar anda. URL dari repo anda ini seharusnya adalah https://github.com/username/praxis-academy:

1. Kerjakan [petunjuk pembuatan repo dari Github ini](https://help.github.com/en/articles/create-a-repo). Cukup kerjakan langkah 1 - 6 saja, setelah itu berhenti pada saat repo telah dibuat. Untuk langkah 5, jangan mengaktifkan checkbox Initialize this repository with a README. Untuk selanjutnya, ini akan kita sebut sebagai repo GitHub.
2. Di komputer lokal, kerjakan (lokasi direktori bebas, asal hak akses memenuhi - bisa create dir, file, dan seterusnya - lokasi yang anda pilih akan disebut sebagai $LOKASI):
   * git clone <https://github.com/username/praxis-academy>
   * untuk selanjutnya, lokasi direktori $LOKASI/praxis-academy ini kita sebut sebagai repo lokal.
3. Latihan dan kasus berikut seharusnya dikerjakan di repo lokal dan hasilnya di-push ke repo Github.

Latihan

1. Untuk keperluan pengerjaan kemampuan dasar ini, buat direktori di repo lokal dengan nama kemampuan-dasar. Setelah itu, masukkan hasil pekerjaan yang telah anda kerjakan pada materi Kemampuan Teknis Dasar 1 di atas ke direktori kemampuan-dasar-1 dalam direktori kemampuan-dasar sehingga pada repo lokal anda akan terdapat 2 direktori. Setelah itu, push ke repo GitHub. Struktur direktori dan berbagai file yang harus ada pada posisi ini adalah (jika menggunakan Linux, pengguna lain silahkan menyesuaikan):

kemampuan-dasar/

└── kemampuan-dasar-1

├── kasus

│   ├── cari-java.sh

│   └── process-killer.sh

└── latihan

├── cmdline.txt

├── env.txt

├── grub.txt

├── kernel-linux.txt

└── shellscripts

├── echo\_example.sh

├── first.sh

....

....

.... seluruh shell script yang dicoba

....

....

└── sleep-example.sh

1. Untuk praktik berikut ini, kerjakan semuanya di dalam direktori kemampuan-dasar-2 di bawah kemampuan-dasar. Cara mengerjakannya adalah dengan mempraktikkan dan kemudian meng-capture / meng-copy hasil dari praktik tersebut ke dalam file [markdown](https://guides.github.com/features/mastering-markdown/). Jika di praktik tersebut diminta untuk membuat repo, buatlah juga di account anda, tidak perlu berada dalam repo praxis-academy. Buat file README.md yang berisi nama-nama repo di account GitHub anda yang berhubungan dengan praktik-praktik tersebut. Praktik-praktik yang harus dikerjakan adalah:
   * [Getting Started](https://dev.acquia.com/blog/getting-started-collaborative-development-git). Nama file hasil untuk praktik ini adalah praktik-1.md.
   * [Team Collaboration with GitHub](https://code.tutsplus.com/articles/team-collaboration-with-github--net-29876). Nama file hasil untuk praktik ini adalah praktik-2.md.
2. Buat direktori wawasan pada direktori kemampuan-dasar. Buatlah rangkuman dari "Ruang Lingkup Pengembangan Sofwtare - Materi dan Penjelasan - point 1 sampa 6" di atas (awal dokumen ini). Tuliskan hasil rangkuman anda pada file README.md, letakkan di direktori wawasan tersebut. Yang anda kerjakan adalah merangkum, bukan menterjemahkan. Pahami esensinya, setelah itu tuliskan secara ringkas.

## Hasil Akhir

Setelah menyelesaikan kemampuan dasar teknis 1 dan 2 di atas, struktur direktori dan file di repo hasil pekerjaan anda (baik repo lokal maupun repo GitHub) adalah sebagai berikut (jika menggunakan Linux, pengguna lain silahkan menyesuaikan):

kemampuan-dasar/

├── kemampuan-dasar-1

│   ├── kasus

│   │   ├── cari-java.sh

│   │   └── process-killer.sh

│   └── latihan

│   ├── cmdline.txt

│   ├── env.txt

│   ├── grub.txt

│   ├── kernel-linux.txt

│   └── shellscripts

│   ├── first.sh

│  ....

│  ....

│  .... seluruh shell script yang dicoba

│  ....

│  ....

│   └── sleep-example.sh

├── kemampuan-dasar-2

│  ├── praktik-1.md

│ ├── praktik-2.md

│  └── README.md

└── wawasan

└── README.md