|  |  |  |
| --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 PEMBANGUNAN APLIKASI | |
| **TAHAP NOSS** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | **CU2/WA5 - COMMIT MODULE SOURCE CODE** | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD 1023 INTRODUCTION TO WEB DEVELOPMENT | |
| **NO DAN TAJUK STANDARD KANDUNGAN KSKV** | K5 BACKUP WEB PROGRAMMING | |
| **NO. KOD NOSS** | IT-010-3:2016-C02/P(68/70) | Muka Surat : 1 Drp : 9 |
| **NO. KOD KSKV** | KPD1023/P(16/17) |

**TAJUK/***TITLE***:**

**SANDARAN KOD PENGATURCARAAN**

**TUJUAN/***PURPOSE* **:**

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

1. Menerangkan tentang penyelenggaraan kod sumber seperti kawalan SCM dan Versi
2. Menerangkan tentang kaedah pengedaran kod sumber *Pull, Push, Check out, Commit, Update*

PENERANGAN:

**5.1** Penyelenggaraan kod sumber

**5.1.1 Source Code Management (SCM)**

1. Juga dikenali sebagai ***source code maintenance*** / ***revision control*** / ***version control***.
2. Merupakan pengurusan terhadap perubahan yang terjadi pada *source code* dan dokumen-dokumen lain semasa proses pembangunan perisian.
3. Dalam proses pembangunan sistem, kumpulan pembangun sistem biasanya terdiri daripada beberapa orang pengaturcara. Oleh yang demikian, *source code* mungkin diubah oleh lebih daripada seorang pengaturcara.
4. Fungsi utamapengurusan kod sumber adalah untuk **menyediakan pengurusan fail dan *version control* supaya ahli pasukan sentiasa peka dengan perubahan yang berlaku, dan hanya versi fail terbaru dikenal pasti untuk digunakan dalam ruang kerja**.
5. Contoh perisian version control:
   * + Git
     + Mercurial
     + Subversion
     + Source Code Control System (SCCS)
     + Vesta
     + AutodeskVault
     + Visual Studio Team Services (VSTS)
     + BitKeeper
6. Terminologi pengurusan dalam SCM:

|  |  |
| --- | --- |
| **Terminologi** | **Fungsi** |
| Source code repository | Merupakan pangkalan data / *server* yang menyimpan semua kod sumber. |
| Branching | Merupakan penduaan objek supaya sebarang perubahan yang dibuat akan berlaku kepada semua objek pada masa yang sama. |
| Revision | Merupakan proses mengurus, merekod dan menjejaki perubahan yang telah dilakukan ke atas sistem / perisian berkenaan.  Proses *revision control* dapat dipermudahkan dengan menggunakan perisian khas yang dikategorikan di bawah ***Version Control System***. |
| Access control | Pendekatan yang digunakan untuk keselamatan *source code.* |
| Distribution | Membolehkan banyak pembangun perisian untuk mengerjakan sesuatu projek yang sama tanpa memerlukan mereka berkongsi rangkaian yang sama |

1. Apakah itu *Source code* dan *Repository*?
   * + - *Source code* merupakan struktur data yang mengandungi metadata bagi fail-fail yang disimpan dalam komputer.
       - *Repository* mengandungi metadata seperti:
         * Sejarah dan rekod perubahan data
         * Objek yang dihantar
         * Rujukan kepada objek yang dihantar

**5.1.2 *Version Control***

1. Version Control adalah sistem yang merekod sebarang perubahan pada fail atau set fail supaya pada bila-bila masa anda boleh kembali ke satu versi fail.

Situasi1 1:

Sebagai contoh, anda akan menggunakan kod sumber perisian sebagai fail kawalan versi, walaupun pada hakikatnya anda boleh melakukan ini pada hampir semua jenis fail pada komputer.

Situasi 2:

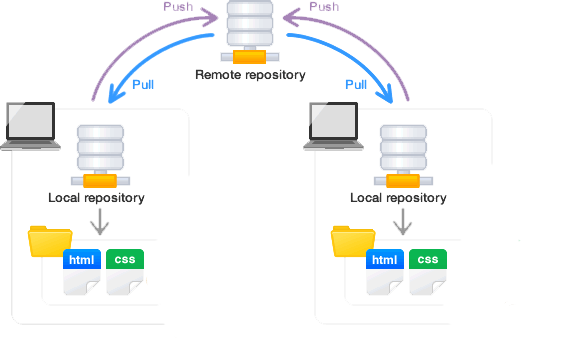
Jika anda seorang pereka grafik atau pereka web dan anda ingin menyimpan setiap versi imej atau susun atur yang anda buat, Version Control adalah penyelesaian yang bijak untuk digunakan.

1. Sistem ini membolehkan anda memulihkan fail anda ke keadaan sebelumnya, memulihkan keseluruhan projek dalam keadaan sebelumnya, dan banyak lagi. Menggunakan VCS bermakna jika anda melakukan kerosakkan pada fail atau kehilangannya, anda dengan mudah boleh mendapatkannya semula.
2. Kebanyakan Version Control melibatkan konsep berikut:-

**Persediaan Asas-**

* **Repositori (repo)**: Pangkalan data menyimpan fail.
* **Server**: Komputer yang menyimpan repo.
* **Client**: Komputer yang menyambungkan ke repo.
* **Set Kerja / Salinan Kerja**: Direktori fail setempat anda, di mana anda membuat perubahan.
* **Trunk / Main**: Lokasi utama untuk kod di repo.

**5.2 Kaedah Pengedaran Kod Sumber (*Source Code Distribution Method*)**

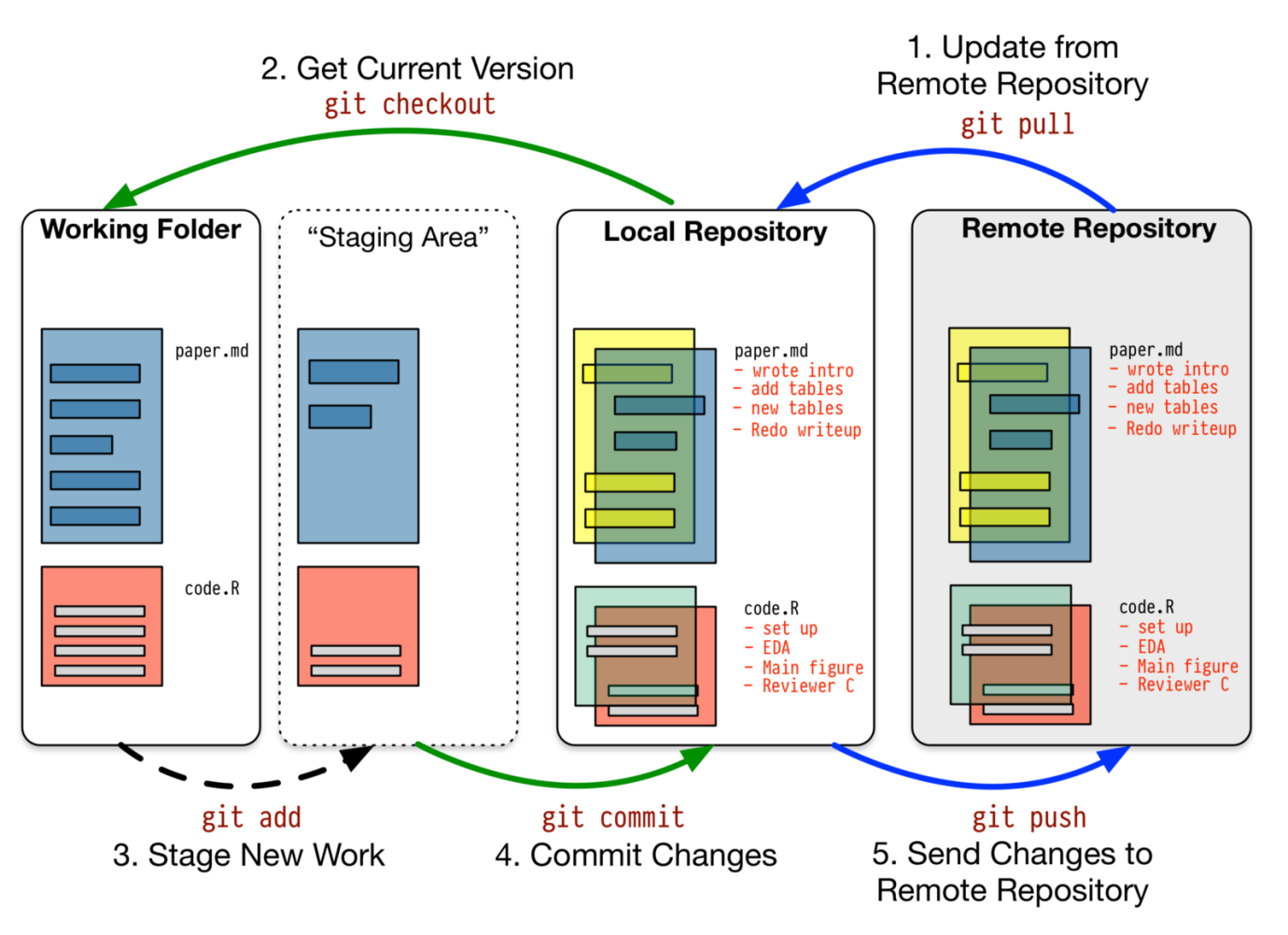


Commit

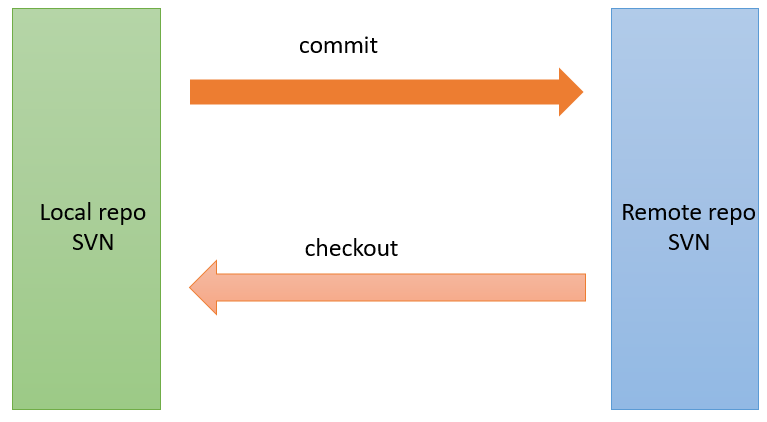
Checkout

**5.2.1 Terminologi :**

* + - **Push**
      * *Upload* / *export* kod sumber dari *local* *repository* ke *repository* utama
      * Terminologi ini digunakan jika terdapat *local* *repository* dan juga *repository* utama
    - **Pull**
      * *Download / import kod sumber dari repository utama ke local repository*
      * *Terminologi ini digunakan jika terdapat local repository dan juga repository* utama
    - **Checkout**
      * Muat turun salinan sumber ke mesin anda
    - **Commit**
      * *Mengubah kod sumber dan menghantar / menyimpan kod sumber yang telah diubah ke repository*
      * *Terminologi ini digunakan jika hanya terdapat satu repository*
    - **Update**
      * *Mengemaskini kod sumber dengan mendapat kod sumber terbaru dari repository*
      * *Terminologi ini* digunakan jika hanya terdapat satu *repository*



Rajah 1: Contoh source code distribution method bagi GIT

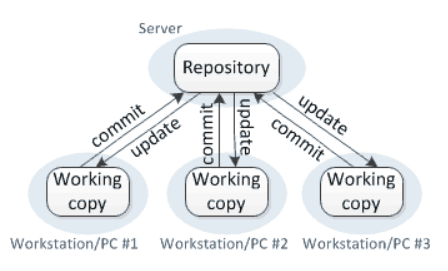


Rajah 2: Contoh source code distribution method bagi SVN

**5.2.2 Dua jenis pengedaran kod sumber:**

* + - **Centralize version control ècontoh: (Subversion)SVN**
    - **Distributed version control è contoh: GitHub**

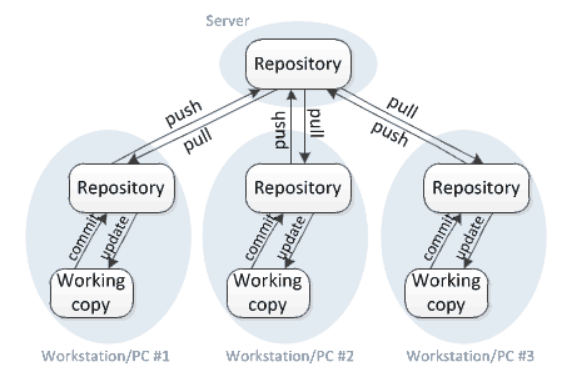
1. *Centralize version control*
   * + Setiap pengaturcara mempunyai *working copy* tersendiri tetapi hanya mempunyai satu *repository* utama.
     + Apabila seorang pengaturcara yang lain *commit* kod sumber ke *repository*, pengaturcara yang lain dapat melihat perubahan kod sumber dan membuat pengemaskinian kod sumber tersebut.



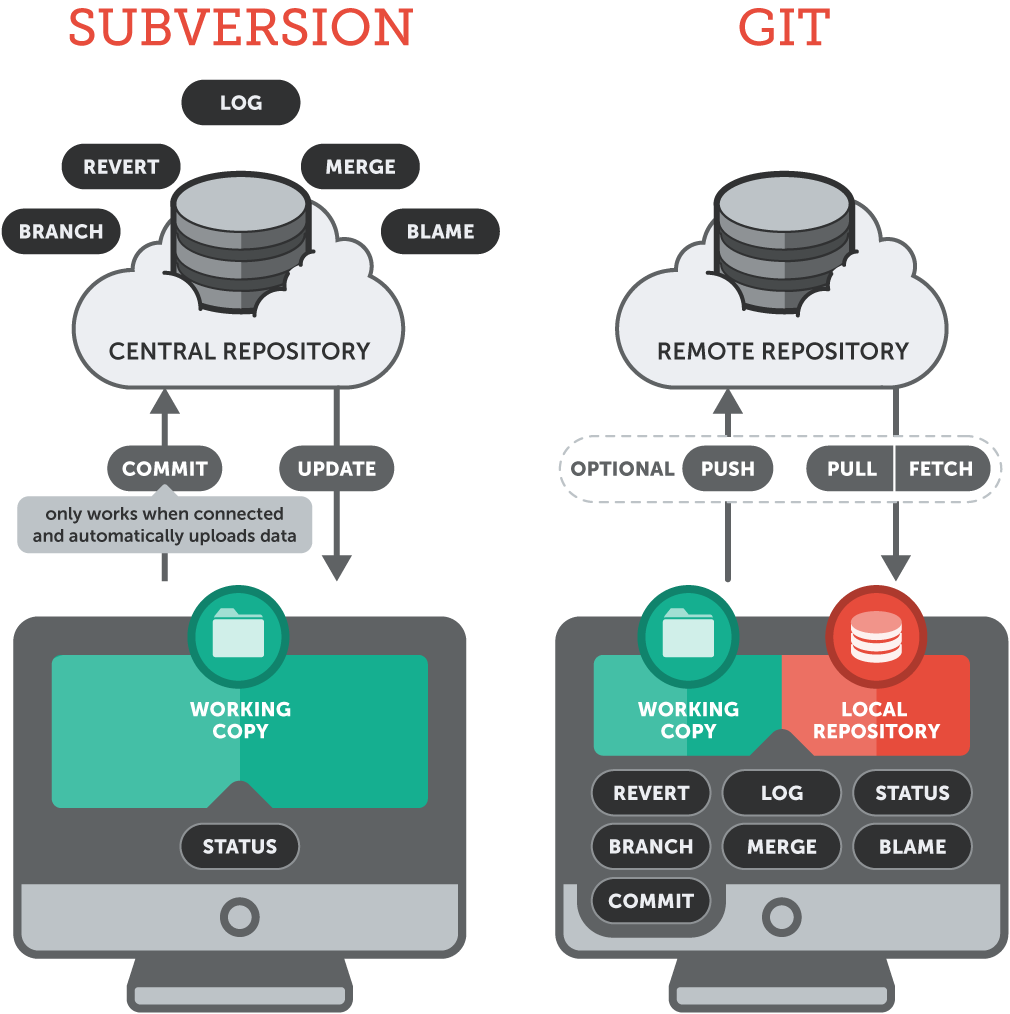
**Rajah 3:** *Centralized Version Control*

1. *Distributed version control*
   * + Setiap pengaturcara mempunyai working copy dan local repository tersendiri.
     + Apabila seorang pengaturcara commit kod sumber yang telah diubah ke local repository, pengaturcara yang lain tidak dapat melihat perubahan tersebut sehingga pengaturcara tersebut *push* kod sumber ke *repository* utama.

**Rajah 4:** *Distributed Version Control*



Perbezaan SVN Dan GIT



SOALAN:

1. Apakah *Source Code Management(SCM)?*
2. Berikan 3 jenis terminologi pengurusan dalam SCM?
3. Senaraikan 3 kaedah pengedaran dalam *Source Code Management(SCM)?*

RUJUKAN:

1. Source Code Management Best Practise

<https://intland.com/blog/sdlc/source-control-management-best-practices/>, 10 Julai 2019, 11.45 a.m

1. Source Code Management

<http://its.unl.edu/bestpractices/source-code-management>, 10 Julai 2019,

1.15 p.m

1. Source Code Management System

<https://www.slideshare.net/xSawyer/source-code-management-systems>,

10 Julai 2019, 3.00 p.m

1. Memulai GIT: Tentang Version Control

<https://git-scm.com/book/id/v1/Memulai-Git-Tentang-Version-Control>,

10 Julai 2019, 5.25 p.m

1. Commit: Version Control

<https://en.wikipedia.org/wiki/Commit_(version_control)>, 11 Julai 2019, 1.15 p.m

1. Distributed: Version Control

<https://en.wikipedia.org/wiki/Distributed_version_control>, 11 Julai 2019,

3.45 p.m

7. Working with GIT:

<https://medium.com/lean-in-women-in-tech-india/working-with-git-9dafa8f986b3>, 12 Julai 2019, 12.45 p.m