|  |  |  |
| --- | --- | --- |
| C:\Users\Admin\Pictures\JATA KPM.png  **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** | 3 | |
| **KOD, NAMA CU DAN WA NOSS** | CU2/WA5 – COMMIT MODULE SOURCE CODE | |
| **KOD DAN NAMA PROGRAM KV** | TEKNOLOGI SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KOD DAN NAMA KURSUS KSKV** | KPD 2023 WEB PROGRAMMING | |
| **NO.DAN TAJUK STANDARD KANDUNGAN KSKV** | K5 BACKUP SOURCE CODE | |
| **NO. KOD NOSS** | IT-010-3:2016-C02/ P(11/11) | Muka Surat : 1 Drp : 7 |
| **NO. KOD KSKV** | KPD 2023 / P(11/11) |

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

COMMIT MODULE SOURCE CODE

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

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

* *Maintenance of source code*
* *Source code distribution method*
* *Format of work progress report*

**PENERANGAN/** *INFORMATION* **:**

**1. Source Code Management (SCM)**

* Adalah sistem atau perisian yang menyediakan koordinasi dan perkhidmatan pengurusan kod sumber diantara ahli-ahli dalam kumpulan pembangunan perisian
* Ia menyediakan pengurusan fail dan kawalan versi supaya tidak berlaku pertindihan atau pengulangan perubahan yang sama ke atas fail kod sumber yang sedang dibangunkan. Setiap ahli hanya akan membuat perubahan fail kod sumber yang terkini
* SCM juga berfungsi ;
* Menyediakan kemudahan bagi pembangun bekerja secara serentak ke atas fail yang sama
* Menggabungkan perubahan-perubahan yang dilakukan oleh pembangun-pembangun lain
* Mengesan dan mengaudit perubahan yang diminta atau yang telah dilakukan
* Mengesan status pembaikan ralat(bug-fix)
* Mengedarkan (*deploy)* versi terkini bagi aplikasi yang dibangunkan

Kebaikan SCM

* Membolehkan pembangun perisian **berkolaborasi** antara satu dengan yang lain tanpa berlaku pertindihan tugas semasa membangunkan kod perisian yang sama
* Membolehkan **keseluruhan sejarah pembangunan suatu perisian dapat dikesan**; perubahan yang dibuat, individu yang bertanggungjawab membuat perubahan
* Berdasarkan perubahan-perubahan yang dikesan SCM boleh **menjana *release notes*** tentang perubahan yang terdapat dalam versi terkini yang diedarkan.
* Membantu dalam menguruskan dokumentasi pembangunanan dan pengujian aplikasi

SCM digunakan oleh :

* Pembangun project ,perisian atau pengatur cara
* Penguji perisian (*tester)*
* Pengguna-pengguna *advanced* yang ingin menguji kod-kod yang belum stabil atau belum dikeluarkan untuk penggunaan

Contoh SCM

* Git
* Subversion (SVN)
* Mercurial
* Bazaar
* VCS

A close up of text on a white background

Description automatically generated2. Persekitaran SCM Git

**Istilah** (berdasarkan SCM *Github*)

* *Working Copy* – salinan fail yang sedang dilakukan perubahan
* *Workspace* – lokasi dalam komputer dimana fail yang sedang dibuat perubahan disimpan contoh di desktop
* *Local Repository* – lokasi fail /code yang disimpan contoh *local server* / *Local Github repository*
* *Remote Repository-* lokasi di github server

**Kaedah Pengedaran Kod Sumber (*Source code distribution method) - Git***

* *Check out* – dapatkan salinan fail yang boleh diedit di *workspace*
* *Check in –* simpan fail ke repository (Commit)
* Commit – simpan / save fail /kod yang telah diubah ke local repository
* *Push –Upload / Export* fail yang telah *commit*  ke *Remote Repository*
* *Pull- Download / import fail / code* dari *remote repository* ke local server
* *Update – mengemaskini fail anda dengan perubahan daripada individu lain*

**3. Kaedah berdasarkan *Version Contro****l* melibatkan konsep berikut:-

* ***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.

**Tindakan Asas**

***Add***: Masukkan fail ke repo untuk kali pertama, iaitu mengesannya dengan Version Control

***Revision***: Apa versi fail pada (v1, v2, v3, dll).

***Head***: Semakan terkini dalam repo.

***Check Out***: Muat turun fail dari repo.

***Check In***: Muat naik fail ke repositori (jika ia telah berubah). Fail mendapat nombor semakan baru, dan orang boleh "menyemak" yang terkini.

***Checkin Message***: Mesej ringkas yang menerangkan apa yang telah diubah.

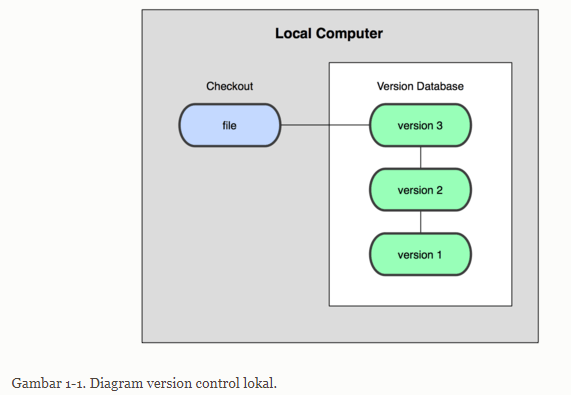
***Changelog / History***: Senarai perubahan yang dibuat pada fail sejak ia dicipta.

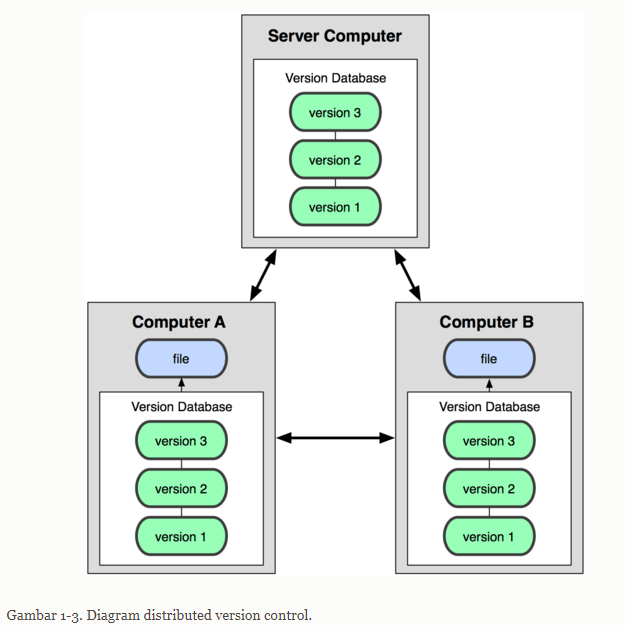
***Update***: Segerakkan fail anda dengan yang terbaru dari repositori. Ini membolehkan anda merebut semakan terkini semua fail.

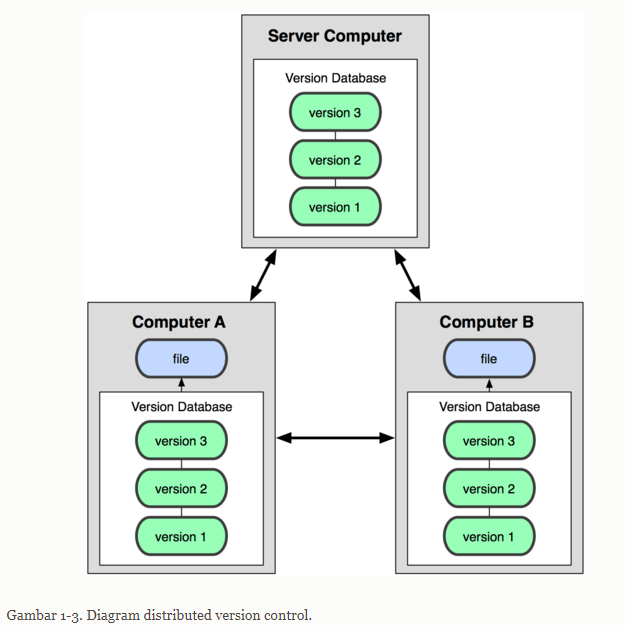
***Revert***: Buang “local changes” anda dan muat semula versi terkini dari repositori.

***Commit***: Dalam sistem version control, commit menambah perubahan terbaru pada kod sumber ke repositori, membuat perubahan ini menukar sebahagian daripada head revision repositori. Tidak seperti commit dalam pengurusan data, version control disimpan dalam repository selama-lamanya. Oleh itu, apabila pengguna lain melakukan kemaskini atau checkout dari repositori, mereka akan menerima versi commit terbaru, kecuali mereka menyatakan mereka ingin mendapatkan versi sebelumnya. Version Control membolehkan kita beralih kembali ke versi sebelumnya dengan mudah. Dalam kontek sini, commit dalam version control dilindungi kerana ia mudah dilancarkan.

**Persekitaran SCM *Version Control***







**4. Laporan kemajuan kerja (*Work progress report)***

i) **maksud** ; adalah dokumen yang melaporkan kemajuan projek bagi tempoh masa tertentu, yang disediakan oleh pekerja untuk dipersembahkan kepada *superior*

ii) Fungsi laporan kemajuan kerja adalah untuk;

a) meyakinkan client atau pelanggan atau *stakeholder* bahawa projek sejang berjalan mengikut perancangan

b) menyediakan maklumat umum tentang dapatan atau progress

c) memberi peluan kepada pelanggan untuk menilai semula projek atau mencadangkan perubahan atau penambahbaikan

d) memberi peluang kepada pelaksana untuk membincangkan masalah yang mungkin timbul semasa projek sedang berjalan

e) pelaksana menyediakan jadual perancangan supaya projek dapat ditamatkan dalam masa yang ditetapkan

iii) Kandungan laporan kemajuan mengandungi ;

a) berapa banyak kerja yang telah disiapkan

b) bahagian kerja atau projek yang sedang berjalan

c) baki atau bahagian kerja yang masih tinggal

d) masalah-masalah yang timbul sepanjang kerja berjalan

e) status projek secara keseluruhan umum

**5.** Contoh borang laporan kemajuan projek

| **Progress / Report** | |
| --- | --- |
| **Project ID:** | [Project ID] |
| **Project:** | [Project Title] |
| **Author:** | [Name], [Job Title] |
| **Date:** | [Generated Date] |

|  |  |  |  |
| --- | --- | --- | --- |
| **Period Covering** |  | **To**: |  |

|  |  |
| --- | --- |
| **Overall Project Status** |  |

|  |
| --- |
| Highlight Progress |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Status** | **Milestone Description** | **Target** | **Latest Forecast** | **Commentary** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**For Reference**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **status** | No information | Will achieve | Will achieve but not be date specified | Will not achieve | ✓ Complete |

**SOALAN /***QUESTION:*

1. Nyatakan fungsi *source code management (SCM) tools.*
2. Berikan beberapa contoh SCM
3. Terangkan kepentingan kod sumber dalam pembangunan aplikasi
4. Terangkan maksud istilah-istilah berikut berdasarkan *SCM Git*

* *Pull*
* *Push*
* *Commit*
* *Update*
* *Check out*

5. Nyatakan persediaan asas bagi *version control*

6. Terangkan kandungan utama dalam sesuatu laporan kemajuan projek

*.*

**RUJUKAN /** *REFERENCE***:**

1. <https://intland.com/blog/sdlc/source-control-management-best-practices/>
2. <http://its.unl.edu/bestpractices/source-code-management>
3. <https://www.slideshare.net/xSawyer/source-code-management-systems>
4. <https://git-scm.com/book/id/v1/Memulai-Git-Tentang-Version-Control>
5. <https://en.wikipedia.org/wiki/Commit_(version_control)>
6. https://en.wikipedia.org/wiki/Distributed\_version\_control
7. http://w3.gel.ulaval.ca/~poussart/gel64324/McMurrey/texte/progrep.htm 10 OGOS 2019 1200
8. https://www.nes.scot.nhs.uk/media/8519/RAG%20Report.doc 16 SEPT 2008 1200