|  |  |  |
| --- | --- | --- |
| logo kpm 2  **BAHAGIAN PENDIDIKAN TEKNIK DAN VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **KERTAS PENERANGAN**  ***(INFORMATION SHEET)*** | | |
| **KOD DAN NAMA NOSS** | IT-010-23 APPLICATION DEVELOPMENT | |
| **TAHAP DAN SEMESTER** | 3 (SEMESTER 3) | |
| **KOD DAN TAJUK KURSUS** | KPD 3033 **DEVELOPMENT ENVIROMENT DEPLOYMENT** | |
| **NO.DAN TAJUK KOMPETENSI** | **K1 DEPLOY SOURCE CODE TO DEVELOPMENT SERVER**  K2 PERFORM DATA POPULATION TO DEVELOPMENT SERVER  K3 PERFORM INTEGRITY SYSTEM TESTING | |
| **NO. KOD KSKV** | KPD 3033 / KP(1/3) | Muka Surat : 1 Drp : |
| **NO. KOD NOSS** | IT-010-3:2016 – C02/ P(1/3) |

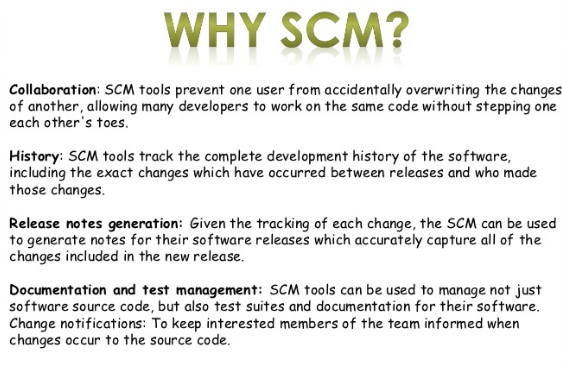
**TAJUK/***TITLE***:**

DEPLOY SOURCE CODE TO DEVELOPMENT SERVER

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

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

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. *Revision control* merupakan proses mengurus, merekod dan menjejaki perubahan yang telah dilakukan ke atas sistem / perisian berkenaan.
   5. **Fungsi utama pengurusan 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**.
   6. Proses *revision control* dapat dipermudahkan dengan menggunakan perisian khas yang dikategorikan di bawah ***Version Control System***.
   7. Contoh perisian version control:
      * Git
      * Mercurial
      * Subversion
      * Source Code Control System (SCCS)
      * Vesta
      * AutodeskVault
      * Visual Studio Team Services (VSTS)
      * BitKeeper
   8. Terminologi dalam SCM:
      * *Source code repository*
        + 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
        + Merupakan pangkalan data / *server* yang menyimpan semua kod sumber.
      * *Working copy* 
        + Fail setempat (*local files*) di mana pengaturcara membuat perubahan pada kod sumber.
        + Terminologi ***check out*** juga digunakan untuk menunjukkan pengaturcara membuat salinan fail daripada *repository* untuk dijadikan *working* *copy*.
      * *Branching*
        + Merupakan penduaan objek supaya sebarang perubahan yang dibuat akan berlaku kepada semua objek pada masa yang sama.



**VERSION CONTROL**

1.9 Apakah version control, dan mengapa anda perlu mengambil berat? 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. 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.

1.10 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. 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.

1.11 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.

**1.12 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 repositori selama-lamanya. Oleh itu, apabila pengguna lain melakukan kemas kini 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 konteks ini, commit dalam version control dilindungi kerana ia mudah dilancarkan.

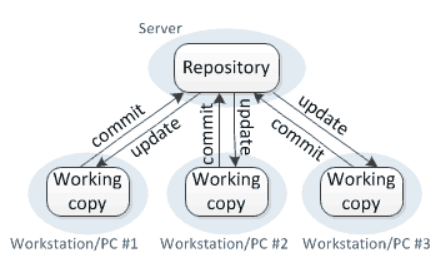
1. **KAEDAH PENGEDARAN KOD SUMBER (*SOURCE CODE DISTRIBUTION METHOD*)**
   1. Terminologi:
      * **Push**
        + *Upload* / *export* kod sumber dari *local* *repository* ke *repository* utama
        + Terminologi ini digunakan jika terdapat *local* *repository* dan juga *repository* utama
      * **Pull**
        + *Download* / *import* kod sumber dari *repository* utama ke *local* *repository*
        + Terminologi ini digunakan jika terdapat *local* *repository* dan juga *repository* utama
      * **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*

2.2 Dua jenis pengedaran kod sumber:

* + - *Centralize version control*
    - *Distributed version control*

2.3 *Centralize version control*

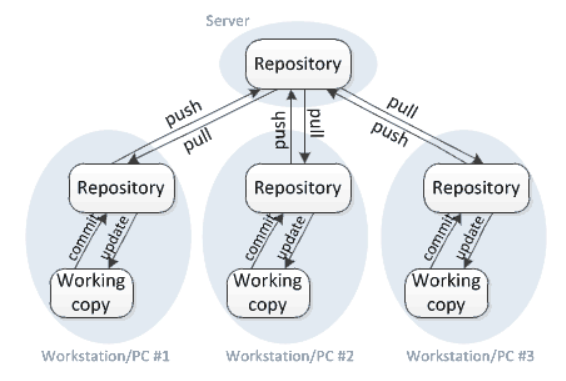
* + - Setiap pengaturcara mempunyai *working copy* tersendiri tetapi hanya mempunyai satu *repository* utama.
    - Apabila seorang pengaturcara yang lain *commit* kod sumber ke *repository*, pengaturcara yang lain dapat melihat perubahan kod sumber dan membuat pengemaskinian kod sumber tersebut.



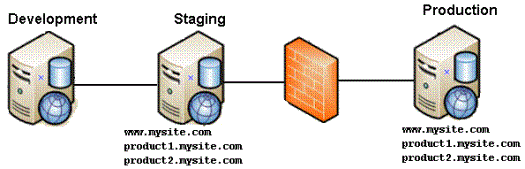
*centralized version control*

2.4 *Distributed version control*

* + - Setiap pengaturcaraan 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.

*distributed version control*

**C1P3 Function of Development Server**

****

**Development Server**

Pelayan pembangunan adalah jenis pelayan yang direka untuk memudahkan pembangunan dan pengujian program, laman web, perisian atau aplikasi untuk pengaturcara perisian. Ia menyediakan persekitaran jangka masa, serta semua utiliti perkakasan / perisian yang penting untuk debugging dan pembangunan program.

Pelayan pembangunan adalah tier teras dalam persekitaran pembangunan perisian, di mana pemaju perisian menguji kod secara langsung. Ia terdiri daripada perkakasan, perisian dan komponen penting yang digunakan untuk menguji dan menguji perisian yang sedang dibangunkan, termasuk penyimpanan pukal, alat platform pembangunan dan utiliti, akses rangkaian dan pemproses tinggi. Setelah selesai ujian, aplikasi itu dipindahkan sama ada ke server pementasan atau pelayan pengeluaran / hidup.

**Testing server**

[**https://helpx.adobe.com/dreamweaver/using/set-testing-server.html**](https://helpx.adobe.com/dreamweaver/using/set-testing-server.html)

**Production Server**

Server pengeluaran adalah sejenis pelayan yang digunakan untuk menyebarkan dan menganjurkan laman web langsung atau aplikasi Web. Ia menganjurkan laman web dan aplikasi Web yang telah menjalani perkembangan dan pengujian yang luas sebelum disahkan sebagai pengeluaran siap. Server pengeluaran juga boleh dirujuk sebagai pelayan langsung.

Server pengeluaran adalah pelayan teras di mana setiap laman web atau aplikasi Web sedang dihoskan dan diakses oleh pengguna. Ia adalah sebahagian daripada keseluruhan perisian dan persekitaran pembangunan aplikasi. Biasanya, persekitaran pelayan pengeluaran, komponen perkakasan dan perisian sama persis dengan server pementasan.

Walaupun, agak terhad kepada penggunaan dalaman seperti dalam pelayan pementasan, pelayan pengeluaran terbuka untuk akses pengguna akhir. Perisian atau aplikasi mestilah diuji dan dibahaskan pada pelayan pementasan sebelum digunakan pada pelayan pengeluaran.

**C1P4 Function of Database**

**C1P5 Module Integration**

**C1P6 Database Administrations**