|  |  |  |  |
| --- | --- | --- | --- |
| **BAHAGIAN PENDIDIKAN DAN LATIHAN TEKNIK VOKASIONAL**  **KEMENTERIAN PENDIDIKAN MALAYSIA**  **ARAS 5 & 6, BLOK E14, KOMPLEKS E,**  **PUSAT PENTADBIRAN KERAJAAN PERSEKUTUAN**  **PELAN MENGAJAR – TEORI**  ***( THEORY LESSON PLAN )***  ***C:\Users\Admin\Pictures\JATA KPM.png*** | | | |
| **KOD DAN NAMA PROGRAM NOSS** | IT-010-3:2016 – APPLICATION DEVELOPMENT | |
| **TAHAP** | C02/WA05 – APPLICATION MODULE DEVELOPMENT | |
| **KOD, NAMA CU DAN WA NOSS** | KPD2033 FUNDAMENTAL OF PROGRAMMING | |
| **NAMA PROGRAM KV** | SISTEM PENGURUSAN PANGKALAN DATA DAN APLIKASI WEB | |
| **KURSUS KSKV** | KPD 2023 WEB PROGRAMMING | |
| **STANDARD**  **KOMPETENSI** | K5 BACKUP SOURCE CODE | | |
| **NO. KOD KSKV** | IT-010-3:2016-C02/P(66/70) | Muka Surat : 1 Drp : 14 | |
| **NO. KOD NOSS** | KPD2033/P(12/17) |

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

**PENYENGARAAN KOD SUMBER**

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

Kertas penerangan ini adalah bertujuan menerangkan mengenai :

* SCM
* Pengedaran Kod Sumber

**SCM**

**SCM adalah satu program yang** **membantu menyimpan semua kod dan setiap perubahan yang pernah anda lakukan**. Selain itu,

(SCM) adalah **perisian yang menyediakan penyelarasan dan perkhidmatan di antara ahli-ahli pasukan pembangunan perisian**. Pada tahap yang paling asas, **ia menyediakan pengurusan fail dan version control** supaya ahli pasukan tidak menulis lebih dari satu sama lain, dan hanya versi fail terbaru dikenal pasti untuk digunakan dalam ruang kerja.

Contoh SCM tools adalah

* **Subversion**
* **Git**
* **Mercurial**
* **Bazar**
* **CVS**

**Sandaran atau biasa disebut sebagai *Backup* merupakan suatu proses merujuk kepada menyalin data yang tersimpan pada storan komputer dari satu lokasi storan ke lokasi storan lain**.Data atau kumpulan maklumat tersebut berbentuk seperti fail, dokumen, gambar, video, audio, sistem komputer, *driver* atau perisian tertentu. Backup biasanya dilakukan dengan tujuan tertentu seperti:

1. Melindungi keselamatan data daripada bencana dan kerosakan storan secara fizikal seperti banjir, kebakaran dan lain-lain
2. Melindungi kerosakan data akibat serangan ancaman digital seperti virus, *malware* atau *spyware*
3. Mengembalikan data dengan mudah sekiranya berlaku kerosakan atau kehilangan data di mana data asal boleh diperolehi daripada media sandaran tanpa perlu membuatnya semula.
4. Mendapatkan data yang lama semula sekiranya diperlukan

*Backup* perlu dilakukan apabila berhadapan dengan situasi berikut:

1. Apabila ada aplikasi komputer baru yang ingin dimasukkan ke dalam server yang sudah mempunyai aplikasi lain di dalamnya
2. Menaiktaraf perisian atau menambah perkakasan ke atas server (*upgrade*)
3. Apabila menaiktaraf atau mengemaskini produk perisian itu sendiri (*update patches*). Proses sandaran perlu dilaksanakan dan di dokumentasikan supaya dapat mengelakkan kehilangan data terkini dan memudahkan kerja *Restore* jika diperlukan
4. Proses sandaran berkala secara berkala samada harian, mingguan atau bulanan bergantung kepada kategori data samada data kritikal atau tidak kritikal.
5. Sebelum menghantar komputer untuk dibaiki atau di format

Bagi produk perisian pula, terdapat 2 kategori *backup* yang perlu diberi perhatian oleh pembangun perisian iaitu ;

1. sandaran terhadap **kod sumber**
2. sandaran terhadap **pangkalan data**

***BACKUP* KOD SUMBER (*SOURCE CODE*)**

**Kod sumber adalah komponen asas program komputer yang dicipta oleh seorang pengaturcara**. Ia boleh dibaca dan mudah difahami oleh manusia. Apabila seorang pengatur cara menaip sebuah urutan teks dalam pernyataan bahasa HTML ke *Windows Notepad*, maka, teks di dalam fail itu dikatakan mengandungi kod sumber. Kod sumber akan menghasilkan sebuah produk perisian seperti sistem berasaskan laman web dan juga aplikasi mudah alih.

Semasa kod sumber dibangunkan, penyimpanan failnya perlu diurus dengan baik. Pembangun perlu peka dengan setiap perubahan terhadap kod sumber. Apalagi jika kod sumber yang melibatkan lebih daripada seorang pembangun. Kod sumber yang telah dilepaskan kepada pengguna jarang melalui proses *backup* oleh pengguna selain salinan yang disimpan oleh pembangun.

***BACKUP* PANGKALAN DATA**

*Backup* terhadap pangkalan data perlu dilakukan dalam beberapa keadaan iaitu

1. Semasa proses pembangunan perisian, pangkalan data yang kosong (tanpa data) perlu disalin supaya dapat dipasang semasa dilepaskan kepada pengguna dengan keadaan yang belum dimasukkan sebarang data. Biasanya pembangun akan memasukkan data *dummy* untuk menguji sistem yang dibangunkan.
2. Pangkalan data yang siap dipasang untuk digunakan oleh pengguna akhir tentunya mengandungi data yang sebenar. Data-data ini penting untuk dilaksanakan proses backup *pada* waktu tertentu supaya dapat dikembalikan semula sekiranya berlaku sebarang kerosakan pada data.
3. Data yang terlalu lama yang tidak perlu dipamerkan lagi kepada pengguna juga perlu diurus dengan baik. Data ini perlu disalin dan disimpan dengan selamat supaya dapat disediakan atau dikembalikan jika diperlukan pada bila-bila masa.

**PENGURUSAN DAN KAWALAN VERSI/KELUARAN(Version Control)**

**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**.Semasa dalam proses pembangunan, kod sumber sentiasa mengalami perubahan demi perubahan sebelum menjadi sebuah produk yang telah siap. Produk perisian yang siap dihasilkan akan diserahkan kepada pengguna untuk digunakan. Bagaimanapun, perisian tersebut kemungkinan besar juga akan mengalami evolusi perubahan setelah sekian lama digunakan. Dari sudut lain pula, pemaju perisian akan sentiasa berusaha menghasilkan produk perisian baru yang lebih berkemampuan untuk menggantikan produk perisian lama. Bagi memudahkan kerja-kerja penyelenggaraan dan penghasilan produk baru, satu skema pengenalan telah digunakan iaitu versi dan keluaran.

**Versi sistem ialah bentuk skema pengenalan, biasanya dalam bentuk nombor berlabel pada sistem**. Sistem baru merupakan sistem yang pertama kali dibina manakala sistem tambahan atau versi merupakan sistem yang telah diubah suai dan menunjukkan fungsian yang lebih baik daripada sistem sebelumnya. Kombinasi beberapa versi boleh diintegrasikan menjadi satu produk yang dikenali sebagai keluaran. Keluaran merupakan skema pengenalan yang dicipta khusus untuk menggambarkan kelainan kombinasi versi antara satu keluaran dengan keluaran yang lain. Versi baru kadang-kadang dicipta untuk kegunaan dalaman dan tidak semestinya dilepaskan kepada pengguna.

Versi-versi baru perlu menunjukkan kemampuan dan pencapaian sistem yang lebih baik daripada yang sebelumnya. Ada kemungkinan versi baru dicipta atas sebab-sebab tertentu seperti membolehkan sistem berfungsi pada persekitaran perisian atau perkakasan yang berbeza, membetulkan ralat sistem dan meningkatkan kelajuan sistem sedia ada.

Dalam aplikasi GITHUB, setiap kod sumber yang diubahsuai oleh individu dianggap versi. GITHUB akan mengawal versi tersebut dengan mengasingkan fail yang diambil oleh individu lain dengan fail yang asal. Setiap tugas dalam pembangunan sistem juga boleh dibahagikan mengikut *branches* atau dedahan iaitu menjadi versi yang kecil. Dedahan-dedahan ini akan digabungkan sekiranya satu-satu tugas itu selesai.

**Cara menyimpan dan mengawal *version* dalam *Google Drive***

Selain menggunakan aplikasi Github terdapat satu cara lain yang boleh dilakukan untuk kita melakukan penyimpanan data atau maklumat yang boleh dikongsi serta di kawal nombor versinya. Berikut adalah merupakan langkah untuk menguruskan fail version menggunakan google drive:

**SENARAI KAEDAH PENGEDARAN KOD SUMBER**

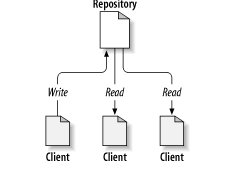
Kod **Sumber**:

**Merujuk kepada sebarang pernyataan yang ditulis dalam bahasa pengaturcaraan komputer yang difahami manusia dan terdapat dalam beberapa fail computer.**

* + - Terminologi: **Pull**
      * *Download / import kod sumber dari repository utama ke local repository*
      * *Terminologi ini digunakan jika terdapat local repository dan juga repository* utama
    - **Push**
      * *Upload* / *export* kod sumber dari *local* *repository* ke *repository* utama
      * Terminologi ini digunakan jika terdapat *local* *repository* dan juga *repository* utama
* **Check out-semak**
* Membuat pertukaran antara versi kod sumber yang tersedia ada yg terdapat dalam local sistem.(switch between versions of code already on the local system.)
  + - **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*

**Repositori**

**Repositori merupakan pusat penyimpanan data**. Repositori **menyimpan informasi dalam bentuk susunan sistem file – hirarki umum dari file dan direktori**. Beberapa *client* tersambung ke repositori, dan kemudian membaca atau menulis ke file ini. Dengan penulisan data, *client* membuat informasi tersedia bagi yang lain dengan membaca data, *client* menerima informasi dari yang lain.



Rajah 1: Sistem Client/Server umum

Rajah 1 merupakan gambaran file server secara umum. Repositori juga dianggap sejenis file server, walaubagaimanapun, ia bukan seperti yang anda bayangkan.

Repositori subversion istimewa kerana ia mengingati setiap perubahan yang pernah ditulis: setiap perubahan ke setiap file dan bahkan perubahan ke susunan direktori itu sendiri seperti penambahan, penghapusan dan pengaturcaraan ulang dari file serta direktori.

Apabila *client* membaca data daripada repositori, secara normalnya ia akan nampak versi terbaharu dalam *filesystem tree.*

Dengan adanya sistem ini, *client* boleh juga untuk *review* *state* terdahulu dalam *filesyste,.*

Sebagai contoh soalan mengenai *history* file: “Apakah yang telah berlaku pada direktori ini pada hari rabu lepas?”

“Siapa yang telah menggunakan dan melakukan perubahan terhadap fail tersebut?”

Jadi, “jantung” VCS adalah direka untuk merekod dan menjejaki perubahan sepanjang masa.

**SOALAN/***QUESTION***:**

1. Apakah SCM.
2. Terangkan konsep Subversion
3. Terangkan konsep repositori
4. Berikan 2 tujuan Sandaran.
5. Berikan Lima jenis pengedaran kod sumber.

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

1. Keith Cooper, Linda Torczon(2011), Engineering a Compiler (2nd Edition), Morgan Kaufmann, ISBN: 978-0120884780
2. Todd Zaki Warfel (2009), Prototyping: A Practitioner's Guide, Louis Rosenfeld Media, LLC, ISBN I-933820-21-7
3. Steve McConnell (2004), Code Complete: A Practical Handbook of Software Construction, (2nd Edition),Microsoft Press, ISBN: 978-0-7356-1967-8
4. Michael L. Scot (2009) , Programming Language Pragmatics, Third Edition, Morgan Kaufmann, ISBN-13: 978-0123745149
5. Simon Marlow (2013), Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming (1st Edition), O'Reilly Media, ISBN-13: 978-1449335946