

Code Review and Security Audit in Private Cloud

Arief Karfianto

sysadmin at UKP4

Arief.Karfianto@ukp.go.id

I. Pendahuluan

Electronic Government (e-Gov) saat ini telah menjadi tren sekaligus kebutuhan dalam pelaksanaan kegiatan pemerintahan. Institusi pemerintah dituntut untuk menerapkan e-Gov dengan memanfaatkan teknologi world-wide-web agar dapat memberi informasi dan layanan kepada masyarakat. Untuk menerapkan e-Gov, kementerian/lembaga membuat berbagai aplikasi berbasis web, baik yang dikerjakan secara mandiri oleh pegawai di dalam organisasi, maupun oleh pihak ketiga. Aspek keamanan aplikasi tersebut menjadi sangat penting karena aplikasi tersebut mengandung informasi harus dijaga kerahasiaan, keutuhan, dan ketersediaannya.

Pada implementasinya, banyak ditemukan aplikasi yang dikembangkan belum memiliki keamanan yang optimal. Apabila celah keamanan tersebut dieksploitasi oleh pihak lain, maka akan berdampak buruk terhadap layanan maupun kredibilitas dari institusi tersebut. Setidaknya terdapat dua sebab sehingga hal tersebut terjadi, pertama disebabkan kurangnya perhatian pengembang (*programmer*) pada keamanan aplikasi yang dibuatnya, dan kedua disebabkan kurangnya evaluasi keamanan (*security audit*) pada aplikasi, sebelum maupun sesudah dirilis.

Security auditor memiliki peran yang penting dalam menjaga keamanan aplikasi yang dikembangkan dan dikelola. Mereka harus memastikan proses pengembangan software dilakukan dengan metode yang aman. Salah satu usaha pengamanan yang dapat dilakukan adalah dengan mengimplementasikan mekanisme *code review* dan *security testing* dalam *development environment* yang identik dengan *production environment*. Kebutuhan tersebut dapat diimplementasikan dengan memanfaatkan private cloud yang dimiliki oleh organisasi.

II. Code Review dan Security Testing

Membuat aplikasi yang aman merupakan tanggung jawab seluruh *stakeholders* yang terlibat dalam proses software development lifecycle (SDLC). Mereka terdiri dari analis (kebutuhan bisnis), desainer/arsitek aplikasi, programmer, tester, dan bagian operasional. Pengembangan juga melibatkan project management serta yang tidak kalah pentingnya adalah tim audit keamanan.

Keamanan aplikasi harus divalidasi melalui *code review* dan *security testing*. *Security testing* dan *code review* sebaiknya dilakukan bersamaan dengan pengujian fungsional dan sebelum aplikasi tersebut dirilis. Hal tersebut dapat dilakukan secara manual ataupun terautomatisasi. Tools untuk melakukan *code review* bukanlah sebagai solusi untuk seluruh isu keamanan, namun hanya untuk melakukan identifikasi pada sebagian kode yang memerlukan perhatian. Seringkali untuk menghindari false positive dan false negative, code review tools seringkali melewatkan beberapa celah keamanan. Oleh sebab itu, selain menggunakan tools, *code review* secara manual juga perlu dilakukan.

Pengujian yang umum dilakukan antara lain dengan menguji kemungkinan *overflow* dan *injection*, serta format input acak yang tidak diharapkan (*fuzz testing*). Selain metode tersebut, terdapat banyak metode untuk melakukan pengujian khususnya untuk *web application security*. Salah satunya adalah OWASP top 10 list. Open Web Application Security Project (OWASP)¹ adalah project open source yang dibangun untuk menemukan penyebab dari tidak amannya sebuah software dan menemukan cara menanganinya. Ada 10 celah keamanan aplikasi web yang ditemukan dan rekomendasi mereka tentang menanganinya sebagai sebuah standard keamanan minimal dari aplikasi web. OWASP top 10 list tahun 2013 adalah sebagai berikut:

1. Injection
2. Broken Authentication and Session Management
3. Cross Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration
6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross Site Request Forgery (CSRF)
9. Using Components with Known Vulnerabilities
10. Unvalidated Redirect and Forwards

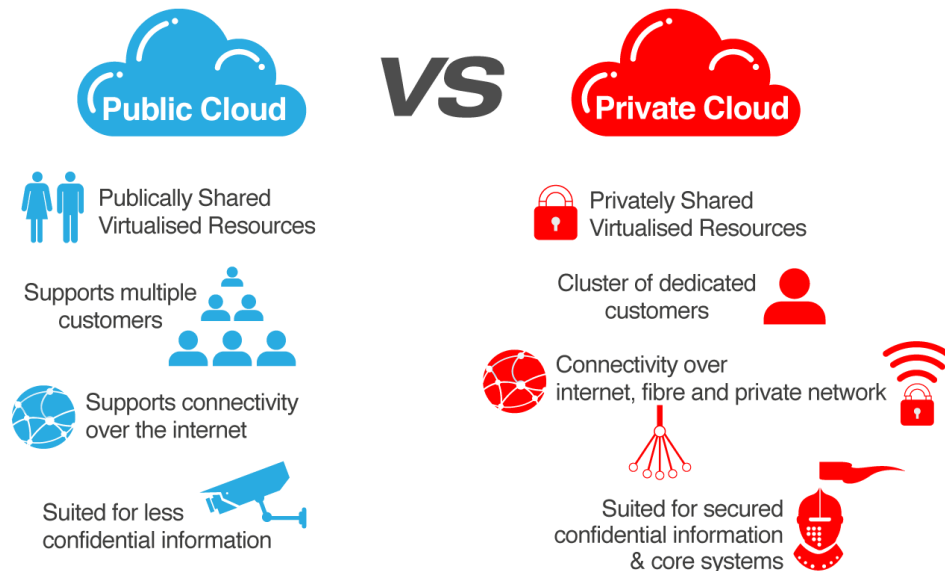
Pengujian dengan contoh kriteria di atas perlu dilakukan pada lingkungan pengembangan yang mensimulasikan konfigurasi pada lingkungan *production*.

III. Private Cloud

Seorang system administrator berperan besar dalam kelangsungan mekanisme tersebut. Pengembang software dan security auditor memerlukan *software environment* tertentu untuk mengembangkan dan menguji code mereka. Environment tersebut harus tersedia dalam waktu singkat, dan sumber dayanya harus dapat digunakan oleh keperluan atau pengembangan aplikasi lainnya karena environment tersebut tidak dibutuhkan selamanya. Fleksibilitas tersebut dapat diperoleh dengan menggunakan *cloud computing*.

¹ <https://www.owasp.org>

Cloud computing telah terbukti menjadi alternatif yang baik bagi suatu organisasi karena mampu mengurangi biaya dan meningkatkan fleksibilitas. Faktor keamanan dan ketersediaan merupakan hal yang sangat diperhatikan, sehingga banyak organisasi memilih private cloud dibandingkan menggunakan public cloud.



Sumber: <http://www.skali.net/>

Sebuah private cloud yang juga disebut internal cloud atau corporate cloud berada di dalam parameter lingkungan organisasi (firewall), aksesnya terbatas, dan biasanya hanya untuk pegawai atau rekan kerja.

Gartner Institute telah mendefinisikan lima atribut utama untuk private cloud:

- Menyediakan sumber daya (infrastruktur dan aplikasi) sebagai layanan
- Memiliki fleksibilitas dan skalabilitas yang dapat disesuaikan dengan permintaan pengguna
- Adanya aktivitas berbagi sumber daya pada jumlah user yang besar
- Pengukuran dan pembayaran didasarkan pada penggunaan layanan
- Penggunaan protokol teknologi internet untuk mengakses sumber daya di cloud

Private cloud dapat diimplementasikan dalam beberapa model antara lain:

1. Infrastruktur sebagai Layanan (*Infrastructure as a service*)
2. Perangkat lunak sebagai layanan (*software as a service*)
3. Sistem pendukung regulasi internal (*self regulation system*)

Pada tulisan ini akan dijelaskan poin yang ketiga, dimana private cloud dimanfaatkan dalam fase pengujian keamanan dan penjaminan kualitas dari pengembangan aplikasi (*secure application development*).

Virtualisasi yang merupakan komponen private cloud ini dapat menggunakan berbagai hipervisor seperti KVM, HyperV, Proxmox, RHEV, OpenVZ, atau VMware.

Fitur yang dimanfaatkan pada infrastruktur cloud ini antara lain:

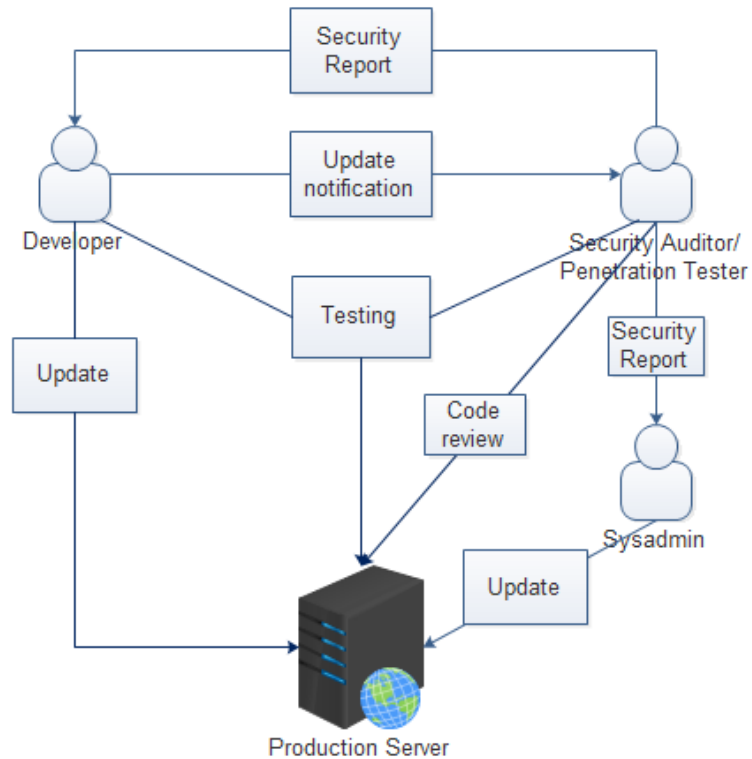
- Flexible Resource
Kemampuan untuk melakukan kostumisasi resource suatu virtual server, antara lain CPU, Memory, Network Interface, Harddisk dan sebagainya.
- Snapshot and Restore
Fitur untuk membuat *image copy* untuk menyimpan data dan konfigurasi (*state*) pada suatu waktu yang kemudian dapat dikembalikan pada kondisi tersebut saat dibutuhkan
- Clone
Fitur untuk membuat salinan yang identik dari suatu *virtual machine*
- High Availability
Layanan high availability ini menjamin ketersediaan (*uptime*) server walaupun host (*physical*) servernya mati. Layanan HA ini akan memindahkan virtual machine ke physical machine yang hidup yang berada dalam satu cluster.

IV. Implementasi *Code Review* dan *Security Testing*

Dalam melakukan security testing, terdapat tiga metode yang dapat digunakan, yaitu

- Blackbox : *tester* melakukan pengujian dengan informasi hanya alamat aplikasi
- Greybox : *tester* melakukan pengujian dengan informasi alamat aplikasi dan credential dengan akses terbatas
- Whitebox : *tester* melakukan pengujian dengan informasi alamat aplikasi, credential, dan akses ke *source code*.









Pada mekanisme yang tidak terkendali, *code review* biasanya dilakukan langsung pada *production server*. Hal ini memiliki konsekuensi developer memiliki akses ke *production server* sehingga menimbulkan risiko terjadinya kebocoran informasi. Selain itu, *security tester* juga melakukan pengujian langsung ke *production server* sehingga ada kemungkinan terjadinya gangguan pada operasional aplikasi. Mekanisme ini tergambar seperti di bawah ini:



Adapun kebutuhan pada sistem yang dibangun pada private cloud ini antara lain:

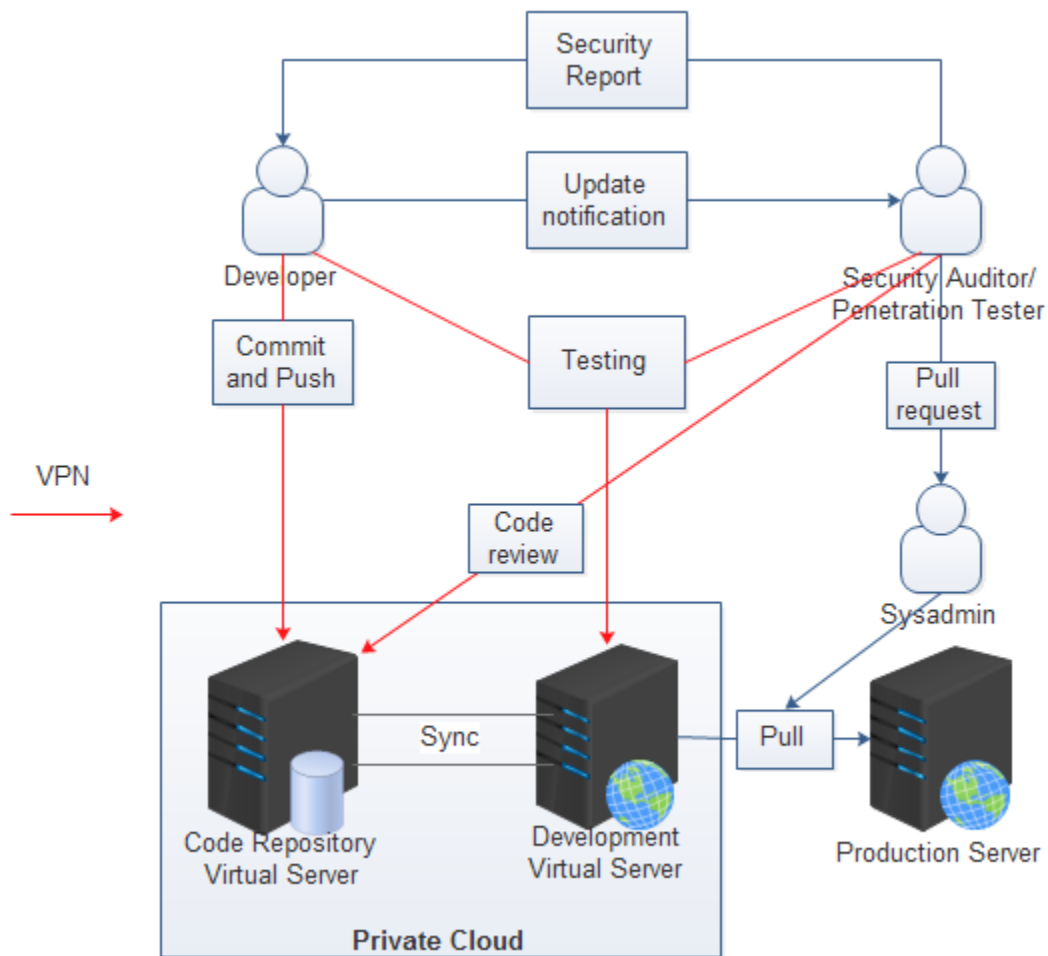
1. Source code harus disimpan dalam sebuah repository
2. Seluruh perubahan source code harus dapat di-review dan memiliki versioning
3. Source code hanya dapat diakses oleh developer yang diberi akses
4. Developer tidak memiliki akses ke lingkungan production
5. Perubahan yang di-commit pada repository dapat dicoba pada lingkungan development
6. Perubahan dapat dilakukan ke production hanya jika telah memenuhi pengujian keamanan

Berikut perbandingan berbagai model security testing:

	Tanpa source control	Dengan source control pada physical server	Dengan source control pada public cloud	Dengan source control pada private cloud
Source code versioning				
Kemudahan dan kecepatan deployment				

Proteksi akses ke production server				
Private access dan management				

Berikutnya akan dijelaskan implementasi pembuatan repository, cloning source code, commit dan push code, dan proses *code review*. Secara garis besar, requirement yang telah disebutkan sebelumnya, digambarkan dalam gambar dibawah ini:



Setelah infrastruktur private cloud telah selesai dibuat, tahap selanjutnya adalah melakukan instalasi sistem. Komponen yang dibangun terdiri dari Code Repository Virtual Server (CODE) dan Development Virtual Server (DEV) dengan komponen sebagai berikut:

	CODE	DEV
Operating System	Linux Server	Linux Server
vCPU	2	2
Memory	4	8
Package	Apache ² Git ³ Gitweb ⁴ Gitis ⁵ OpenSSH ⁶	Apache PHP ⁷ MySQL ⁸ Git OpenSSH

1. Konfigurasi code repository virtual server oleh sysadmin

Dalam membuat sebuah repository, penulis membuat bash script sederhana bernama `addrepo.sh`.

```
#!/bin/bash
echo "Please enter repository name and description"
read -e -p "Name :" -n 30 reponame
read -e -p "Description :" -n 30 repodesc
echo "Creating a repository..."
mkdir /srv/repos/git/$reponame
chown -R git:git /srv/repos/git/$reponame
cd /srv/repos/git/$reponame
git init
git add .
git commit -m 'Initial upload of the project'
cd ..
git clone --bare --shared $reponame/.git/ repositories/$reponame.git
chown -R git:git /srv/repos/git/repositories/$reponame.git/
echo "#!/bin/sh" >> /srv/repos/git/repositories/$reponame.git/hooks/post-receive
echo "GIT_WORK_TREE=/srv/repos/git/$reponame git checkout -f" >>
/srv/repos/git/repositories/$reponame.git/hooks/post-receive
chmod +x /srv/repos/git/repositories/$reponame.git/hooks/post-receive
echo $repodesc > /srv/repos/git/repositories/$reponame.git/description
ln -s /srv/repos/git/repositories/$reponame.git/ /var/cache/git/$reponame.git
echo "[Done]"
```

Berikut adalah contoh penggunaan script `addrepo.sh` untuk membuat repository :

```
root@revision-control ~# ./addrepo.sh
Please enter repository name and description
Name :sample-app2
```

² <http://www.apache.org/>

³ <http://git-scm.com>

⁴ <https://git.wiki.kernel.org/index.php/Gitweb>

⁵ <http://git-scm.com/book/en/Git-on-the-Server-Gitis>

⁶ <http://www.openssh.com/>

⁷ <http://php.net/>

⁸ <http://www.mysql.com/>

```

Description :Sample application 2.0
Creating a repository...
Initialized empty Git repository in /srv/repos/git/sample-app2/.git/
# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
Cloning into bare repository repositories/sample-app2.git...
done.
warning: You appear to have cloned an empty repository.
[Done]

```

Hasil dari script tersebut akan membuat:

- Repository dalam direktori /srv/repos/git/
- Bare repository dalam direktori /srv/repos/git/repository/
- Link pada /var/cache/git/ yang dapat diakses melalui web browser

Selanjutnya sysadmin membuat pengaturan pada repository agar dapat diakses oleh programmer yang diberi hak akses. Sysadmin kemudian menambahkan public key orang yang diberi akses dalam folder gitosis-admin/keydir. Buka file gitosis-admin\gitosis.conf dan tambahkan baris berikut ini:

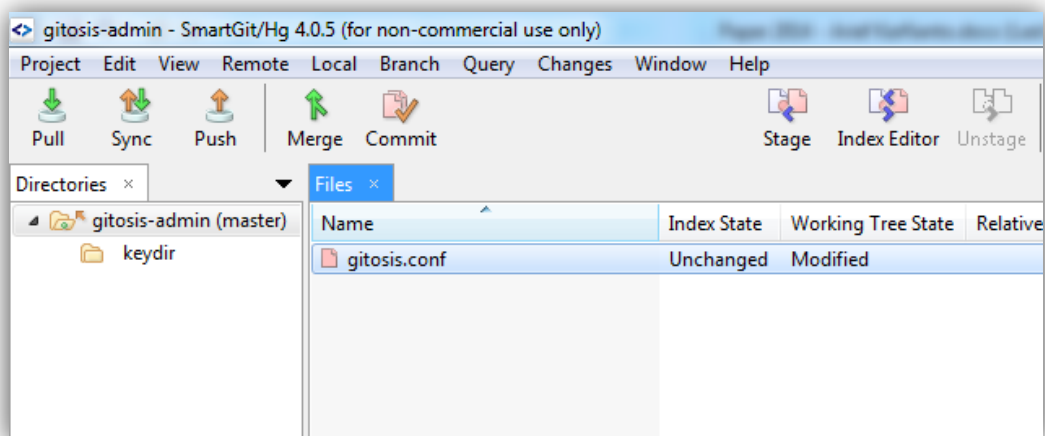
```

[group sample-app2]
writable = sample-app2
members = intruder@LENOVOY460 John@Doe.PC

```

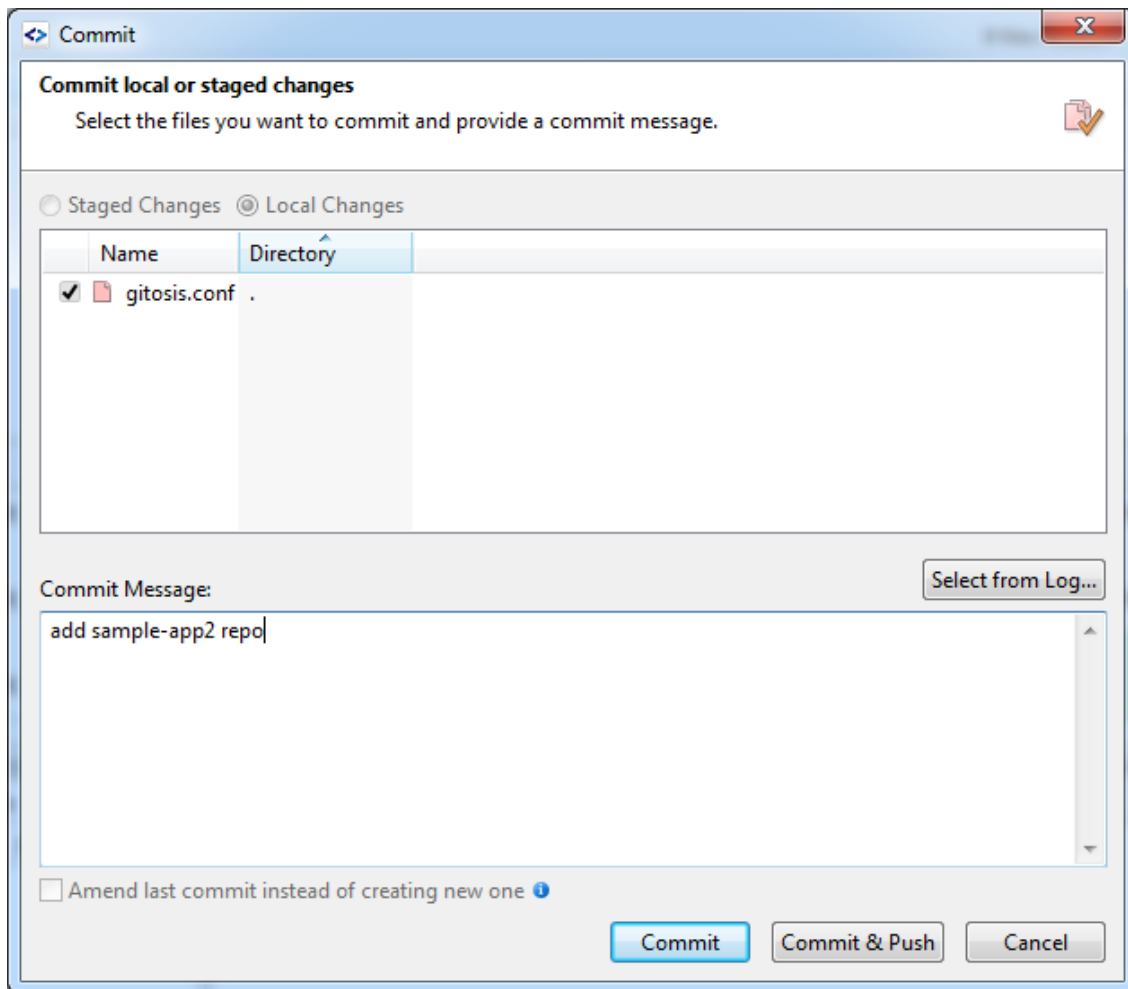
Lalu commit perubahan tersebut ke code server

Buka Smartgit⁹ → Project Gitosis-admin

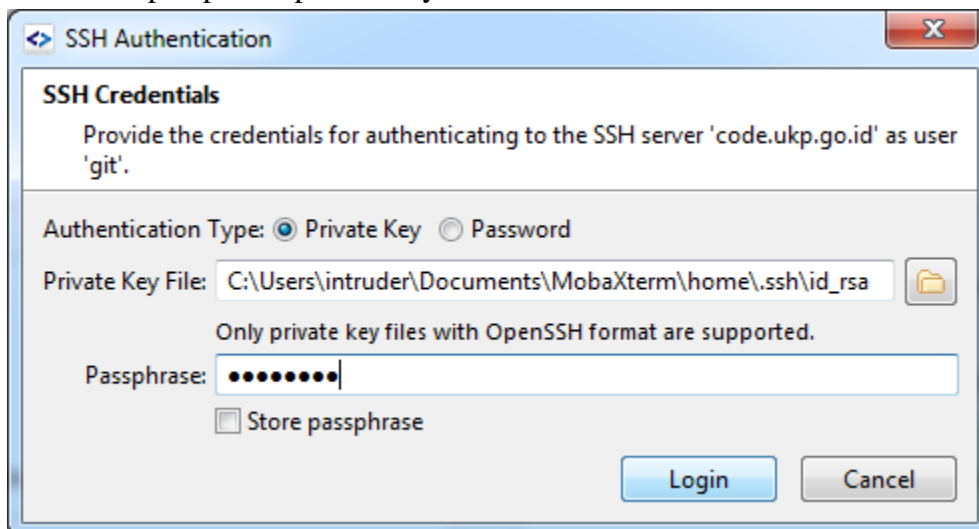


Klik Commit, tambahkan commit message lalu klik commit dan push

⁹ <http://www.syntevo.com/smartgit/>

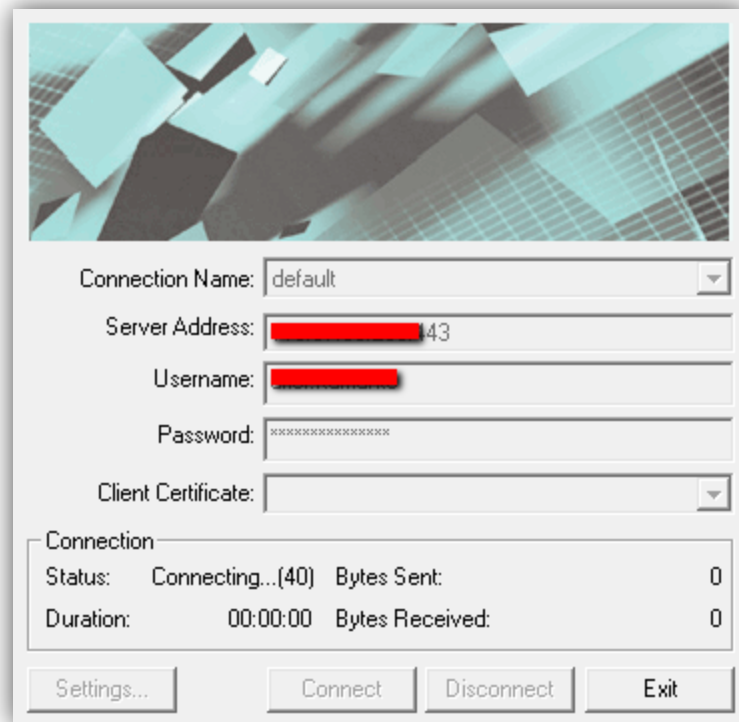


Masukkan passphrase private key.

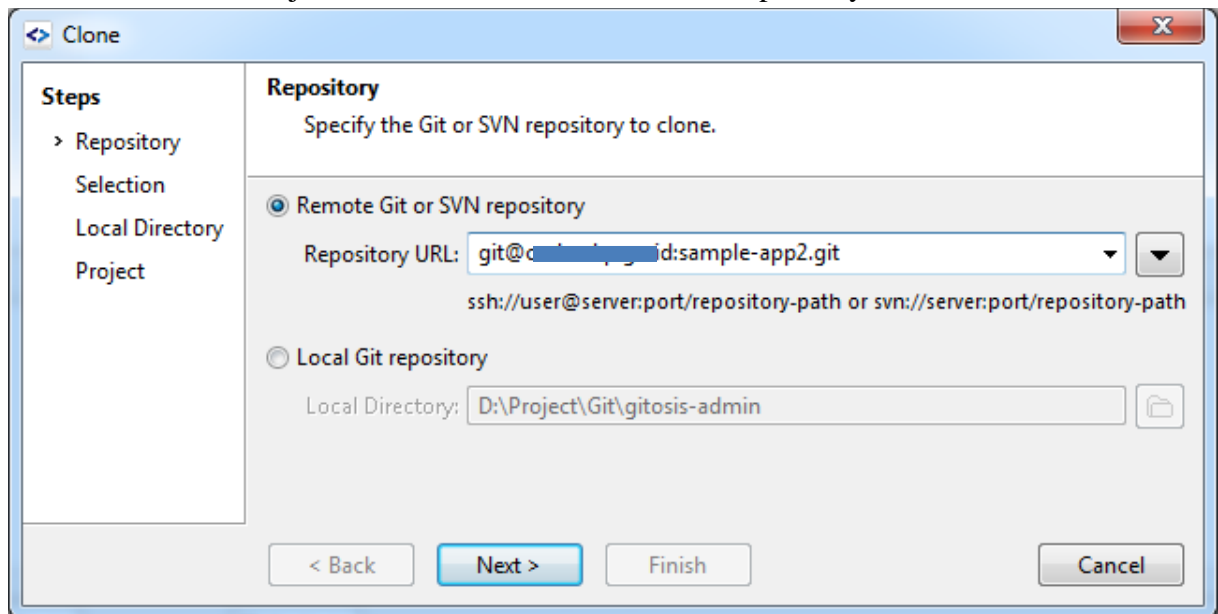


2. Cloning repository ke komputer client oleh developer

Untuk melakukan cloning source code dari repository, programmer harus terkoneksi via VPN ke jaringan *private cloud*.



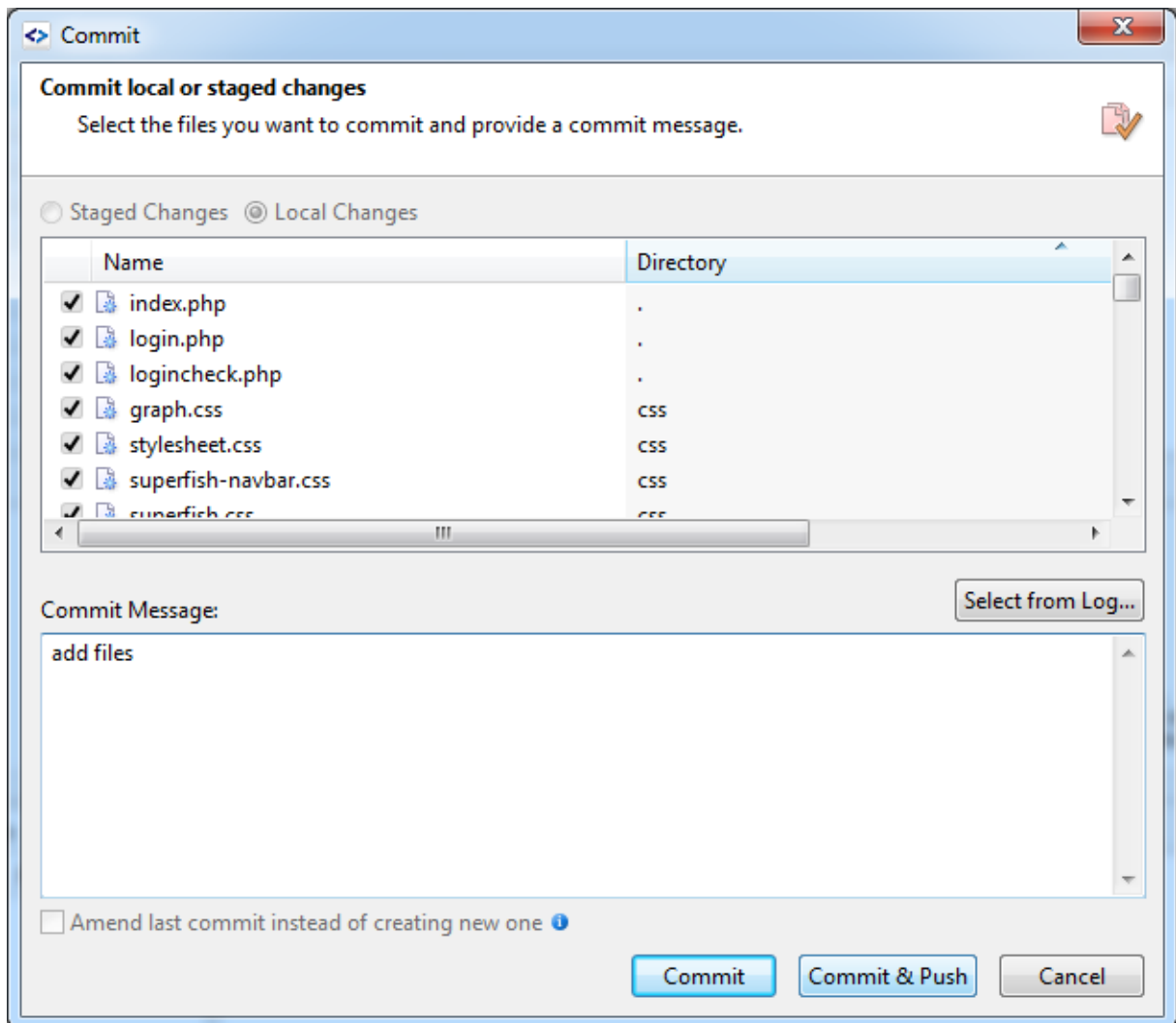
Buka SmartGit → Project → Clone lalu masukkan URL repository



Masukkan nama folder lalu klik finish.

3. Penambahan code pada repository oleh developer

Saat developer menambahkan file ke dalam local repository, maka file tersebut akan tampil dengan status untracked. Untuk membuat perubahan, developer melakukan commit dan push.



4. Code review oleh security tester

Untuk melakukan *code review*, *security tester* mengakses gitweb dan membuka repository yang akan di-review. Pada interface ini terdapat fitur search untuk mencari commit atau fungsi yang diinginkan.

projects / sample-app2.git / blob

commit ▼ ? search: ☐ re

[summary](#) | [shortlog](#) | [log](#) | [commit](#) | [commitdiff](#) | [tree history](#) | [raw](#) | [HEAD](#)

[add files](#)

[sample-app2.git] / index.php

```
1 <?php
2
3 require "inc/pagestart_top.php";
4
5 // if the user can only access one application, redirect to it.
6 if ($_SESSION['num_apps'] == 1) {
7     $host = $_SERVER['HTTP_HOST'];
8     $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
9     $dir = $_SESSION['app_dir']['oneapp'];
10    $protocol = (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] != 'off' ||
11    header("Location: $protocol$host$uri/$dir");
12 }
13 ?>
14 <!DOCTYPE HTML>
15 <html>
16 <head>
17 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
18 <title>Applications</title>
```

Saat terjadi *update* pada source code, security tester dapat melakukan evaluasi pada perubahan yang terjadi.

```
diff --git a/index.php b/index.php

index 5f2dd0c...a07fd04 100644 (file)

--- a/index.php
+++ b/index.php
@@ -7,7 +7,7 @@ if ($_SESSION['num_apps'] == 1) {
     $host = $_SERVER['HTTP_HOST'];
     $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
     $dir = $_SESSION['app_dir']['oneapp'];
-    $protocol = (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] != 'off' || $_SERVER['SERVER_PORT'] == 443) ? "https://" : "http://";
+    $protocol = "http://";
     header("Location: $protocol$host$uri/$dir");
 }
?>
```

IV. **Simpulan**

Dengan implementasi *code review* dan *security testing*, maka aplikasi yang dikembangkan akan lebih mudah untuk dievaluasi dan akses ke source code akan lebih terkendali. Selain itu, *tracking* terhadap *bug* dalam aplikasi akan lebih mudah untuk diidentifikasi dengan melihat history perubahan pada *repository*.

Selain itu, penggunaan private cloud, lingkungan development akan lebih mudah dibuat dan aksesnya terlindungi karena diakses melalui *secure link* (VPN).