Gambar 3-7. Pemberian notifikasi kesalahan secara manual oleh *integrator*

|  |
| --- |
| Penjelasan:  *Developer* Y berperan sebagai *integrator* yang mengeksekusi *build* pada mesin integrasi tanpa menggunakan bantuan *automated build tool*. Apabila hasil dari eksekusi *build* yang dilakukan oleh *integrator* gagal, maka *integrator* mengirimkan notifikasi kesalahan kepada setiap *developer* untuk segera diperbaiki. |

Gambar 3.8**.**  Pengarsipan paket aplikasi secara manual oleh *integrator*

|  |
| --- |
| Penjelasan:  *Developer* Y berperan sebagai *integrator* yang mengeksekusi *build* pada mesin integrasi tanpa menggunakan bantuan *automated build tool*. Hasil dari setiap eksekusi *build* yang berhasil dilakukan oleh *integrator* menghasilkan paket aplikasi. Setiap paket aplikasi tersebut akan diarsipkan oleh *integrator* pada mesin integrasi. |

Gambar 3-9. Pembuatan *report* kemajuan proses pembangunan perangkat lunak oleh *integrator*

|  |
| --- |
| Penjelasan:  *Developer* Y yang berperan sebagai *integrator* yang membuat *report* kemajuan proses pembangunan P/L tanpa *automated continuous integration*. Hasil dari *report* tersebut akan diinformasikan oleh *integrator* kepada setiap *developer*. |

Gambar 3-17. Eksekusi *build* pada mesin lokal

|  |
| --- |
| Penjelasan:  *Developer X* mengeksekusi *build script* dengan bantuan *automated build tool.* Sebelumnya, *developer* X membuat modul A beserta kode pengujian unit dan integrasi modul A di *repository* lokal. Kemudian *developer* X memanggil *target* pada *build* *script* melalui *automated build tool*. *Automated build* akan mengotomasi *target* pengujian kode program dan penyimpanan versi dari modul yang sudah diubah secara otomatis. Umumnya *build script* yang digunakan oleh setiap *developer* adalah sama. Penyamaan *target* dilakukan tim untuk menyamakan alur kerja dari setiap *developer*. |

Gambar 3-18. Eksekusi *build* pada mesin integrasi

|  |
| --- |
| Penjelasan:  *Developer* Y berperan sebagai *integrator* akan mengeksekusi *build script* dengan *automated build tool*. *Build* *script* tersebut akan mengotomasi *target* pengujian kode program dan pembuatan paket aplikasi pada mesin integrasi. Sebelum mengeksekusi *build* *script*, *integrator* harus mengambil (*pull*) kode program dari *repository* pusat dengan menggunakan VSC *tool*. Sebaiknya, *build script* pada mesin lokal dibuat terpisah dengan *build script* pada mesin integrasi, kerena adanya kemungkinan anggota tim menggunakan IDE yang berbeda. |

Gambar 3-20. Penjadwalan eksekusi *build script* pada mesin integrasi

|  |
| --- |
| Penjelasan:  *Automated CI tool* mengeksekusi *build script* dengan menggunakan *automated build tool*. Penjadwalan dapat dilakukan setiap malam (*nightly build*) pada mesin integrasi. Ketika waktu penjadwalan telah tiba, maka *automated* CI *tool* akan mengambil (*pull*) kode program dari *repository* pusat dan mengeksekusi *build script*. |

Gambar 3-21. Notifikasi kesalahan secara otomatis dari mesin integrasi

|  |
| --- |
| Penjelasan:  Pengiriman notifikasi hasil *build* dilakukan dengan bantuan *automated* CI *tool*. Notifikasi hasil *build* yang dieksekusi oleh *automated build tool* akan dikirimkan oleh *automated* CI *tool* kepada setiap *developer* apabila hasil *build* gagal. |

Gambar 3-22. Pengarsipan paket aplikasi oleh mesin integrasi secara otomatis

|  |
| --- |
| Penjelasan:  *Automated CI* *tool* mengeksekusi *build script* melalui *automated* *build* *tool*. Kemudian *automated build tool* mengeksekusi proses *build* secara otomatis. Paket aplikasi yang dihasilkan dari eksekusi *build script* akan diarsipkan oleh *automated build tool* dan disimpan dalam mesin integrasi. |

Gambar 3-23. *Report* kemajuan proses pembangunan perangkat lunak secara otomatis

|  |
| --- |
| Penjelasan:  *Automated CI tool* akan membuat *report* kemajuan proses pembangunan perangkat lunak. *Report* tersebut disimpan pada mesin integrasi. Setiap *developer* dapat melihat kemajuan proses pembangunan lunak tersebut. |