

BAB V

PENUTUP

5.1 Kesimpulan

Setelah tim melakukan praktik CI dengan studi kasus medrecapp dapat disimpulkan bahwa CI memiliki manfaat sebagai berikut:

1. Pengurangan resiko pembangunan perangkat lunak.
2. Pengurangan proses manual yang berulang.
3. Visibilitas proyek yang lebih baik.

Peningkatan kepercayaan diri anggota tim terhadap produk perangkat lunak. Manfaat tersebut dicapai dengan menggunakan prosedur, teknik dan *toolset* yang mendukung CI. Adapun prosedur, teknik dan *toolset* yang digunakan dalam pengimplementasian CI adalah:

1. Melakukan kesepakatan awal sebelum melakukan CI, seperti aturan modul dan standar modul yang ingin dibangun.
2. *Tool VCS* yang sesuai kebutuhan anggota tim. Karena anggota tim bekerja secara terdistribusi, maka dipilih *tool* Git untuk mengelola versi kode program. Sedangkan *software hosting* yang digunakan adalah Github.
3. *Tool testing* yang digunakan untuk tingkatan unit dan integrasi. Pada tingkatan unit, pengujian dilakukan langsung oleh anggota tim dengan JUnit. Sedangkan pengujian integrasi dilakukan oleh anggota tim dengan menggunakan FEST.
4. *Tool build* yang digunakan adalah Ant, proses *build* akan dilakukan setiap kali anggota tim melakukan integrasi kode program.
5. *Tool CI* yang digunakan adalah Jenkins. *Tool* tersebut berperan untuk melakukan pengaturan alamat penyimpanan versi, *script build*, menentukan *trigger build*, yaitu ketika melakukan *push* ke Github dan melakukan *poll* SCM setiap satu jam sekali. Sehingga ketika Jenkins memeriksa Github dan menemukan perubahan, maka *build* akan dijalankan.

Tim juga memperoleh kesimpulan tentang perbandingan jumlah proses pembangunan medrecapp yang dilakukan antara implementasi praktik *continuous integration* tanpa *toolset* dan dengan *toolset* (*automated continuous integration*). Perbandingan tersebut didasarkan pada kegiatan pembangunan medrecapp (lihat tabel 5-1)–.

Tabel 5-1. Perbandingan jumlah kegiatan praktik CI tanpa *toolset* dan *automated continuous integration*

No.	Kegiatan	Praktik CI tanpa <i>toolset</i>		Praktik <i>automated continuous integration</i>	
		Jumlah proses	Urutan proses	Jumlah proses	Urutan proses
1.	Penyimpanan versi <i>package</i>	5	<ol style="list-style-type: none"> 1. <i>Developer</i> menduplikasi <i>package</i> di mesin lokal <i>developer</i>. 2. <i>Developer</i> menyimpan detil informasi perubahan versi <i>package</i>. 3. <i>Developer</i> mengambil duplikasi versi <i>package</i> di direktori pusat. 4. <i>Developer</i> menggabungkan versi <i>package</i> dari direktori pusat ke direktori lokal. 5. <i>Developer</i> menyimpan hasil duplikasi versi <i>package</i> ke direktori pusat. 	4	<ol style="list-style-type: none"> 1. <i>Developer</i> tidak menduplikasi <i>package</i>, semua versi disimpan di repository dengan Git. 2. <i>Developer</i> mengelola versi <i>package</i> di repository dengan Git. 3. Git menyimpan informasi detil dari setiap perubahan versi <i>package</i>. 4. Git menggabungkan versi <i>package</i> dari repository pusat ke repository lokal secara otomatis.
2.	Pengujian kode program	5	<ol style="list-style-type: none"> 1. <i>Developer</i> membuat <i>driver</i> pengujian di setiap kode pengujian. 2. <i>Developer</i> mengeksekusi <i>driver</i> pengujian unit satu per satu. 3. <i>Developer</i> mensimulasikan skenario salah dan benar terhadap GUI <i>package</i> secara manual dan berulang kali. 	5	<ol style="list-style-type: none"> 1. <i>Developer</i> membuat <i>test suite</i> dengan JUnit untuk menguji kode program. 2. <i>Developer</i> dapat mengeksekusi lebih dari satu <i>driver</i> pengujian dengan <i>test suite</i> dari JUnit. 3. <i>Developer</i> membuat kode pengujian terhadap GUI <i>package</i> dan

No.	Kegiatan	Praktik CI tanpa <i>toolset</i>		Praktik <i>automated continuous integration</i>	
		Jumlah proses	Urutan proses	Jumlah proses	Urutan proses
			4. <i>Developer</i> yang menguji <i>package</i> tingkat atas, perlu menguji satu per satu <i>package</i> yang ada pada tingkat bawah, termasuk GUI <i>package</i> . 5. Menguji integrasi dari keseluruhan <i>package</i> .		mengotomasi simulasi pengujian tersebut berdasarkan kode pengujian GUI dengan FEST. 4. <i>Developer</i> dapat menguji lebih dari satu GUI dengan <i>test suite</i> dari JUnit. 5. <i>Developer</i> dapat menguji semua <i>package</i> tingkat bawah dengan satu kali eksekusi <i>test suite</i> dari JUnit.
3.	Build aplikasi medrecapp	3	1. <i>Developer</i> mengeksekusi semua <i>driver</i> pengujian. 2. <i>Developer</i> menguji GUI pada semua <i>package</i> . 3. <i>Developer</i> membuat paket aplikasi .	3	1. Ant mengotomasi eksekusi semua <i>driver</i> pengujian. 2. Ant mengotomasi eksekusi pengujian GUI pada semua <i>package</i> . 3. Ant mengotomasi pembuatan <i>package</i> di mesin integrasi.
4.	Pengintegrasian <i>package</i>	3	1. Integrator memberikan notifikasi kesalahan kepada para <i>Developer</i> . 2. Integrator mengarsipkan paket aplikasi di mesin integrasi. 3. <i>Integrator</i> membuat <i>report</i> kemajuan proses pembangunan aplikasi medrecapp di mesin integrasi	4	1. Jenkins mengeksekusi build di mesin integrasi berdasarkan penjadwalan 2. Jenkins mengotomasi pemberian notifikasi kesalahan kepada para developer 3. Jenkins mengotomasi pengarsipan paket aplikasi medrecapp di mesin integrasi. 4. Jenkins mengotomasi pembuatan

No.	Kegiatan	Praktik CI tanpa <i>toolset</i>		Praktik <i>automated continuous integration</i>	
		Jumlah proses	Urutan proses	Jumlah proses	Urutan proses
					report kemajuan proses pembangunan aplikasi medrecapp.

5.2 Saran

Tugas Akhir ini masih belum sempurna. Maka ada baiknya jika dapat dilanjutkan atau digunakan sebagai referensi untuk mengetahui metode membangun perangkat lunak. *Continuous Integration* merupakan metode sangat baik untuk menghasilkan perangkat lunak berkualitas dalam pembangunan perangkat lunak bagi tim pengembang.

Continuous Integration dapat dilakukan dengan teknik dan *tool* yang berbeda dari Tugas Akhir ini. Pembaca dapat mencari referensinya di Buku *Continuous Integration* (karya Marthin Fowler) atau *internet*. Perlu diingat juga sebelum melakukan praktik CI, maka anggota tim harus mempersiapkan *version control system*, *automated testing* dan *automated build*.