# BAB IV
# PENERAPAN KONSEP PEMBANGUNAN PERANGKAT LUNAK SECARA MANUAL DAN *CONTINUOUS INTEGRATION* PADA STUDI KASUS APLIKASI MEDRECAPP

Pada bab ini akan dijelaskan tentang penerapan strategi pembangunan perangkat lunak yang dilakukan secara manual dan *continuous integration* berdasarkan konsep umum pada bab tiga. Penerapan strategi tersebut dilakukan untuk menunjukkan perbedaan proses antara praktik pembangunan perangkat lunak yang dilakukan secara manual dan *continuous integration*. Pada pembangunan perangkat lunak secara manual, strategi yang akan diterapkan mencakup penyimpanan versi, pengujian kode program, dan pengeksekusian *build*. Sedangkan pembangunan perangkat lunak yang dilakukan dengan metode *continuous integration* mencakup penerapan strategi dan penggunaan *tool* dari *version control system*, *automated testing*, *automated build*, dan praktik *continuous integration*.

Studi kasus yang digunakan untuk menerapkan strategi pembangunan perangkat lunak secara manual dan *continuous integration* adalah aplikasi rekam medis berbasis *java desktop*, yang bernama MedRecApp. Pembangunan aplikasi tersebut dilakukan oleh satu tim yang terdiri dari tiga orang *developer*, yaitu Fachrul, Herna, dan Yuanita.

## 4.1 Modul aplikasi MedRecApp

Pembangunan aplikasi MedRecApp dilakukan tim secara terpisah. Mereka membagi pekerjaannya berdasarkan modul-modul yang ada pada aplikasi tersebut. Modul aplikasi MedRecApp dibuat berdasarkan rancangan *database* yang terdapat pada lampiran.

[GAMBAR]

**Gambar 4-1.** Modul pada aplikasi MedRecApp

Pada **gambar 4-1** dijelaskan bahwa aplikasi MedRecApp memiliki empat modul utama, yaitu modul *master*, modul *front office* (pendaftaran), modul poliklinik, dan modul rekam medis. Modul *front office*, modul poliklinik, dan modul rekam medis, memiliki dependensi terhadap modul *master*, sehingga untuk melakukan perubahan data pada modul-modul tersebut dibutuhkan keutuhan data pada modul *master* terlebih dahulu.

Modul *master* terdiri dari lima submodul, yaitu modul spesialis, modul jaminan, modul obat, modul tindakan, dan modul SDM. Pada modul SDM, terdapat tiga submodul yaitu modul dokter, modul perawat, dan modul staf. Modul perawat, modul dokter, dan modul tindakan, memiliki dependensi terhadap modul spesialis, sehingga untuk melakukan perubahan data pada ketiga modul tersebut dibutuhkan keutuhan data pada modul spesialis terlebih dahulu.

Berdasarkan banyaknya modul yang memiliki dependensi terhadap modul *master*, tim akan mengintegrasikan modul dengan metode *bottom-up*. Dengan metode tersebut, setiap *developer* akan membuat modul-modul yang menjadi dependensi terhadap modul lain terlebih dahulu, sehingga para *developer* tidak perlu membuat *stubs* pada saat pengujian integrasi. Urutan dari pengintegrasian modul dengan metode *bottom-up* pada kasus aplikasi MedRecApp dapat dilihat pada **gambar 4-2**.

[GAMBAR]

**Gambar 4-2.** Urutan pengintegrasian modul pada aplikasi MedRecApp

Pada penerapan strategi pembangunan perangkat lunak secara manual dan *continuous integration*, modul yang akan dijadikan sebagai kasus adalah modul spesialis, modul jaminan, dan modul obat. Modul spesialis akan dikerjakan oleh Fachrul, modul jaminan akan dikerjakan oleh Herna, sedangkan modul obat akan dikerjakan oleh Yuanita.

[GAMBAR]

**Gambar 4-3.** Pembagian modul aplikasi MedRecApp yang dilakukan tim

## 4.2 Pembangunan aplikasi MedRecApp secara manual

Setelah tim melakukan pembagian pekerjaan berdasarkan modul aplikasi, mereka akan membuat modul yang berisi kode-kode program pada komputer lokal masing-masing. Modul yang berisi kode-kode program tersebut akan diuji terlebih dahulu oleh setiap *developer* sebelum digabungkan menjadi sebuah modul yang lebih besar. Pengujian tersebut dilakukan setiap *developer* untuk meminimalisasi kesalahan yang mungkin terjadi pada kode program yang telah mereka buat. Setelah kode program lulus dari pengujian, setiap *developer* akan membuat sebuah *folder* baru untuk menyimpan versi dari perubahan kode program tersebut.

*Developer* yang akan menggabungkan modul hasil pekerjaan *developer* yang lain adalah Fachrul. *Folder-folder* versi kode progam dari Herna dan Yuanita akan dikirimkan ke Fachrul untuk digabungkan menjadi modul yang lebih besar. Untuk memastikan hasil penggabungan kode program tersebut minim dari kesalahan, Fachrul akan melakukan pengujian terhadap hasil penggabungan kode program. Setelah Fachrul melakukan pengujian dan memastikan hasil penggabungan kode program telah minim dari kesalahan, Fachrul akan melakukan pengeksekusian *build* untuk mendapatkan paket aplikasi. Setiap paket aplikai hasil eksekusi *build* akan disimpan oleh Fachrul pada *folder* baru. Penyimpanan paket aplikasi tersebut dilakukan Fachrul agar dapat melakukan *rollback* terhadap paket aplikasi. Aktifitas tersebut dilakukan tim secara manual dan berulang kali hingga menjadi satu kesatuan modul aplikasi MedRecApp.

Paket aplikasi yang telah berisi satu kesatuan modul aplikasi akan di-*deploy* ke *customer*. Fachrul melakukan *deploy* aplikasi dengan meng-*copy* secara manual ke *customer*. Sebelum Fachrul men-*deploy* paket aplikasi ke *customer*, Fachrul akan menguji paket aplikasi terlebih dahulu berdasarkan *requirement customer*. Pengujian tersebut dilakukan untuk memastikan bahwa paket aplikasi tersebut minim dari kesalahan.

#### 4.2.1 Penyimpanan versi

#### 4.2.2 Pengujian kode program

#### 4.2.3 Pengeksekusian *build*

### 4.3 Pembangunan aplikasi MedRecApp dengan *continuous integration*

[PROLOG]

#### 4.3.1 *Version control system*

#### 4.3.2 *Automated testing*

#### 4.3.3 *Automated build*

#### 4.3.4 Praktik *continuous integration*