Tujuan utama pengembangan database adalah membuat agar banyak *user* dapat mengakses sebuah data secara bersamaan di waktu yang sama. Akses tidak akan menjadi masalah apabila *user* hanya membaca data tersebut, namun apabila user tersebut juga mengubah data, maka akan terjadi data yang tidak konsisten (*inconsistency data*).

Inconsistency data dapat diatasi dengan suatu mekanisme yang mengatur jalannya transaksi pengaksesan data yang sama tersebut, yaitu concurrency control. Concurrency control adalah proses pengaturan operasi–operasi dalam banyak transaksi yang berjalan secara simultan pada database tanpa mengganggu operasi pada transaksi lainnya sehingga dapat menghasilkan data yang konsisten (Connolly, 2005, p577 ).

*Uncommitted Dependency Problem* (*Dirty Read*)

Contoh dari masalah ini adalah A memiliki saldo sebesar Rp 100.000, kemudian A melakukan transaksi untuk menambah saldonya sebesar Rp 50.000, namun A membatalkan transaksi tersebut sehingga yang seharusnya saldonya Rp 150.000, kembali menjadi Rp 100.000. Pada waktu yang sama, B melakukan transaksi transfer sebesar Rp 20.000 kepada A dan sistem masih membaca saldo baru A (Rp 150.000), sehingga saldo A menjadi keliru Rp 170.000, yang seharusnya hanya Rp 70.000. Saldo yang dibaca oleh B disebut dengan *dirty data* atau *dirty read*. Transaksi akan gagal karena transaksi B membaca dan menambah data yang salah. Ini dikarenakan sistem berasumsi A berhasil melakukan transaksinya, namun tidak dengan pembatalannya karena B sudah bertransaksi terlebih dahulu. Masalah ini dapat diatasi dengan mebuat B tidak dapat bertransaksi terlebih dahulu sampai transaksi yang dilakukan A benar-benar selesai sampai ke pembatalan dan keputusan akhir sudah terbuat.

Source: http://binus.ac.id/malang/2017/09/concurrency-control-pada-database/