



# Rekayasa Perangkat Lunak

DEPARTEMEN PENDIDIKAN NASIONAL  
UNIVERSITAS PENDIDIKAN INDONESIA  
2008



# *Latar Belakang*

- Kemajuan pesat perangkat keras
- Kemajuan dalam teknik-teknik pembuatan perangkat lunak
- Tuntutan yang lebih tinggi terhadap jumlah perangkat lunak
- Tuntutan yang lebih tinggi terhadap mutu perangkat lunak
- Meningkatnya peran pemeliharaan

# Perangkat Lunak (S/W)

**Perangkat Lunak (Software)** adalah source code pada suatu program atau sistem. Perangkat lunak tidak hanya dokumentasi terhadap source code tapi juga dokumentasi terhadap sesuatu yang dibutuhkan selama pengembangan, instalasi, penggunaan dan pemeliharaan sebuah sistem.

*Engineering* atau Rekayasa adalah aplikasi terhadap pendekatan sistematis yang berdasar atas ilmu pengetahuan dan matematis serta aplikasi tentang produksi terhadap struktur, mesin, produk, proses atau sistem.

**RPL** adalah pendekatan sistematis dan matematis untuk membangun, memelihara dan mengubah perangkat lunak

# ***Mutu perangkat lunak***

## Sponsor

Seseorang atau organisasi yang membiayai/membayar selama pengembangan atau perantaraan sistem software dan biasanya mempunyai respon terhadap pengembangan sistem software itu sendiri dengan melibatkan perhitungan biaya yang optimal.

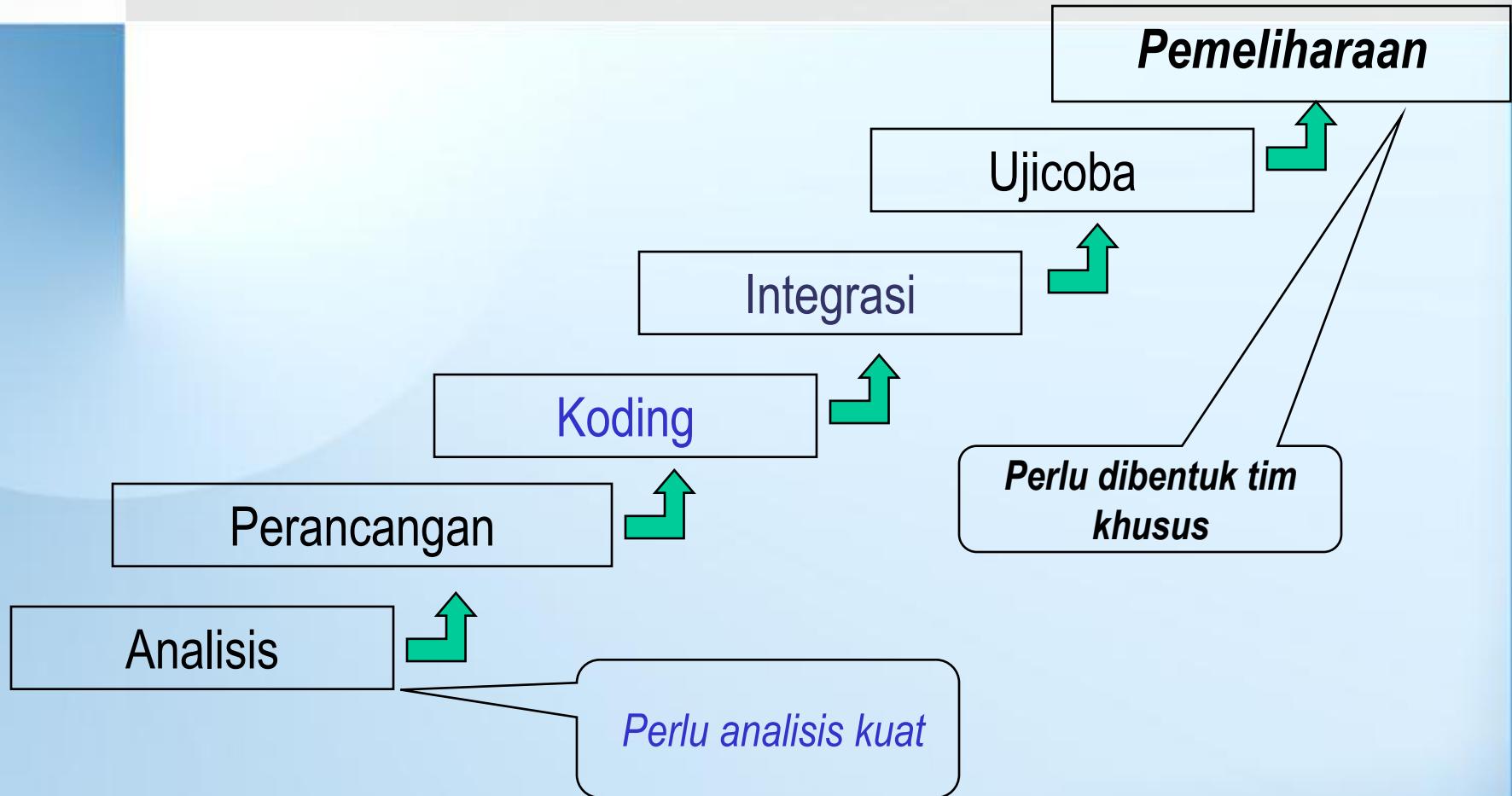
## User

Setiap orang yang secara langsung berinteraksi terhadap eksekusi software, yang secara langsung memberi input ke komputer dan menggunakan/menikmati output dari komputer.

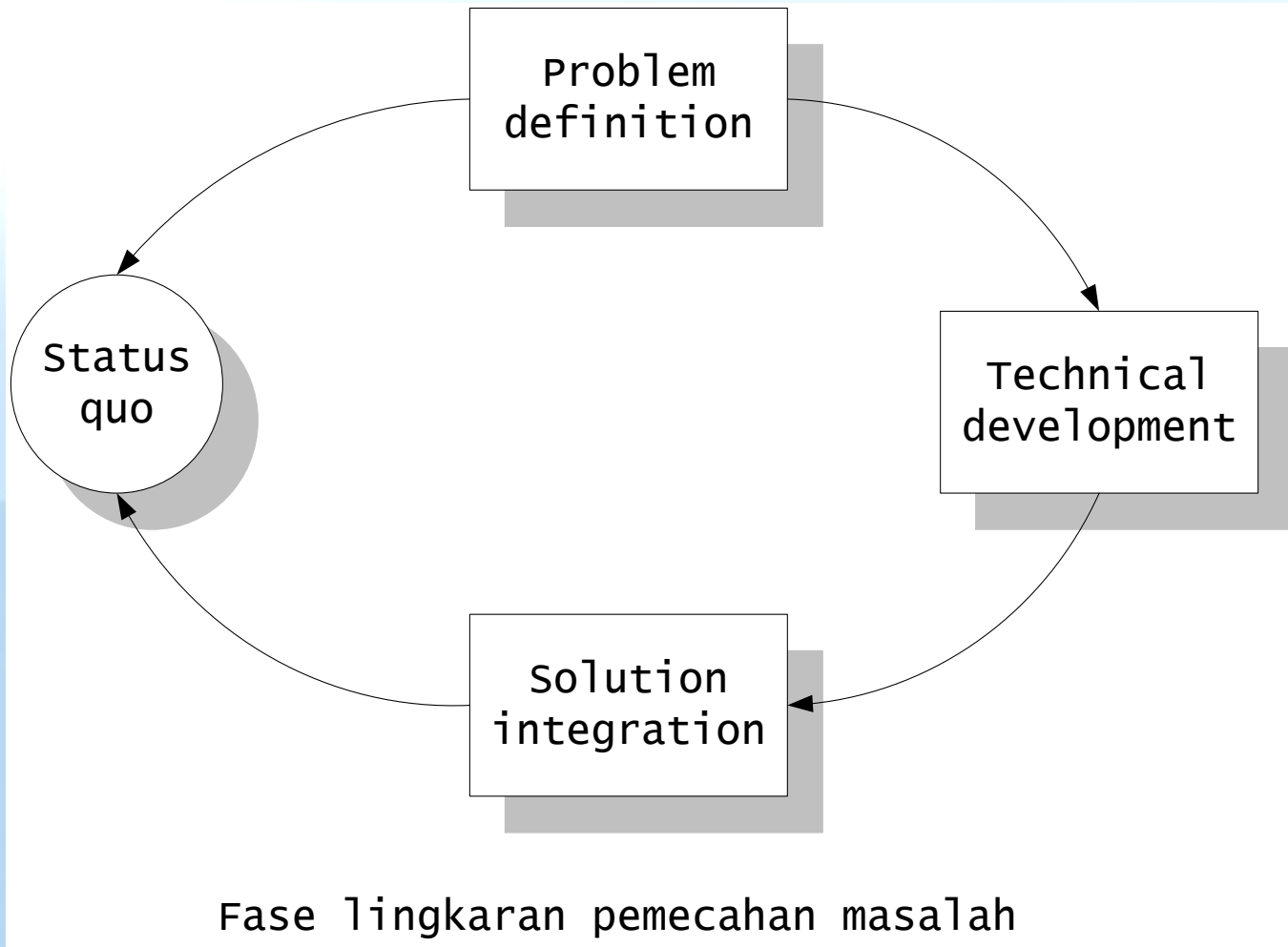
## Developer

Seseorang atau organisasi yang memberikan modifikasi dan memelihara terhadap error serta mengembangkan sistem software tersebut.

# Prinsip-Prinsip RPL

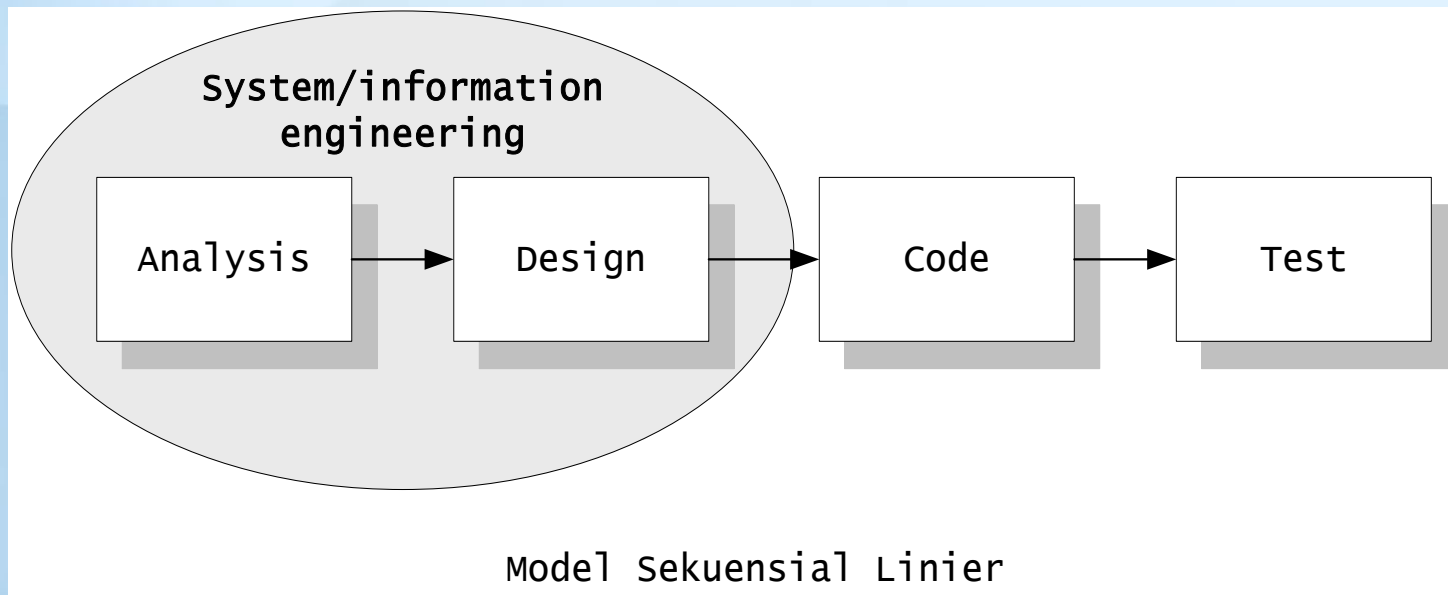


# MODEL-MODEL PROSES PERANGKAT LUNAK



# 1. MODEL SEKUENSIAL LINIER

Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan system pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

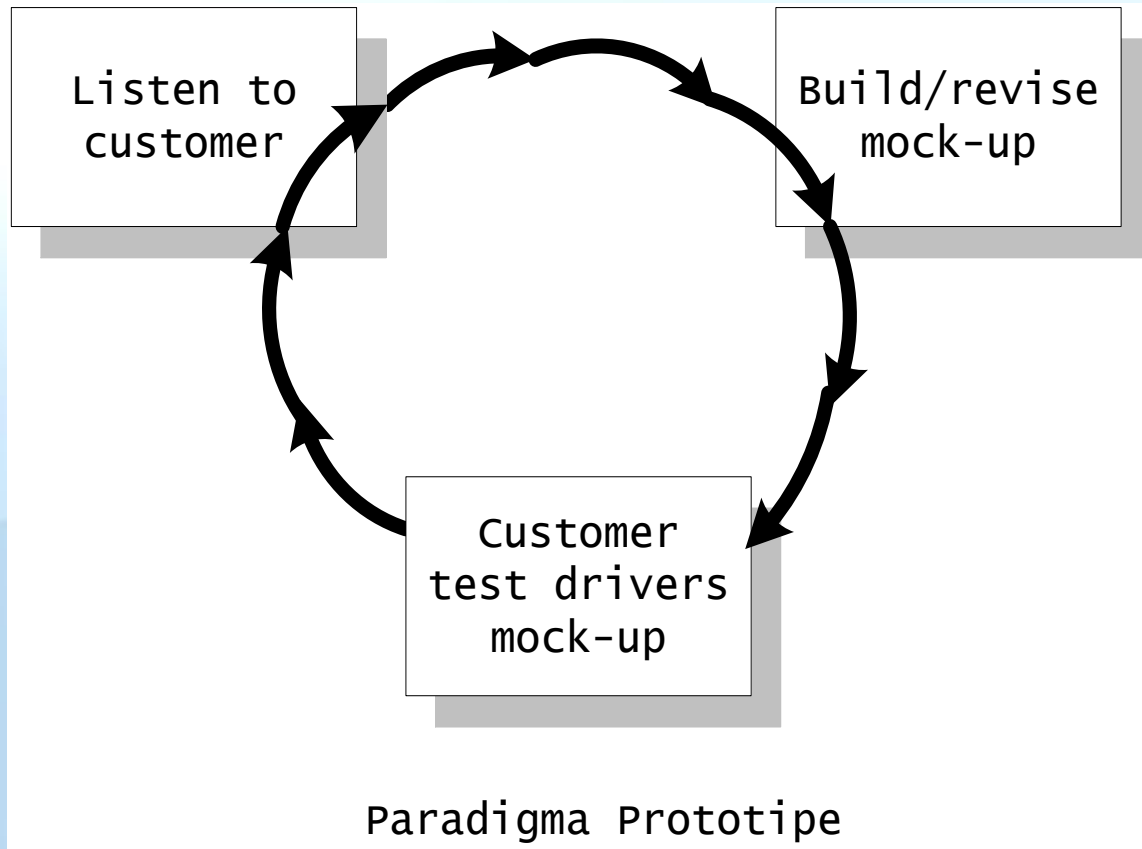


## 2. *MODEL PROTOTIPE*

Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar di mana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”.

Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan/pemakai.

## 2. *MODEL PROTOTYPE*

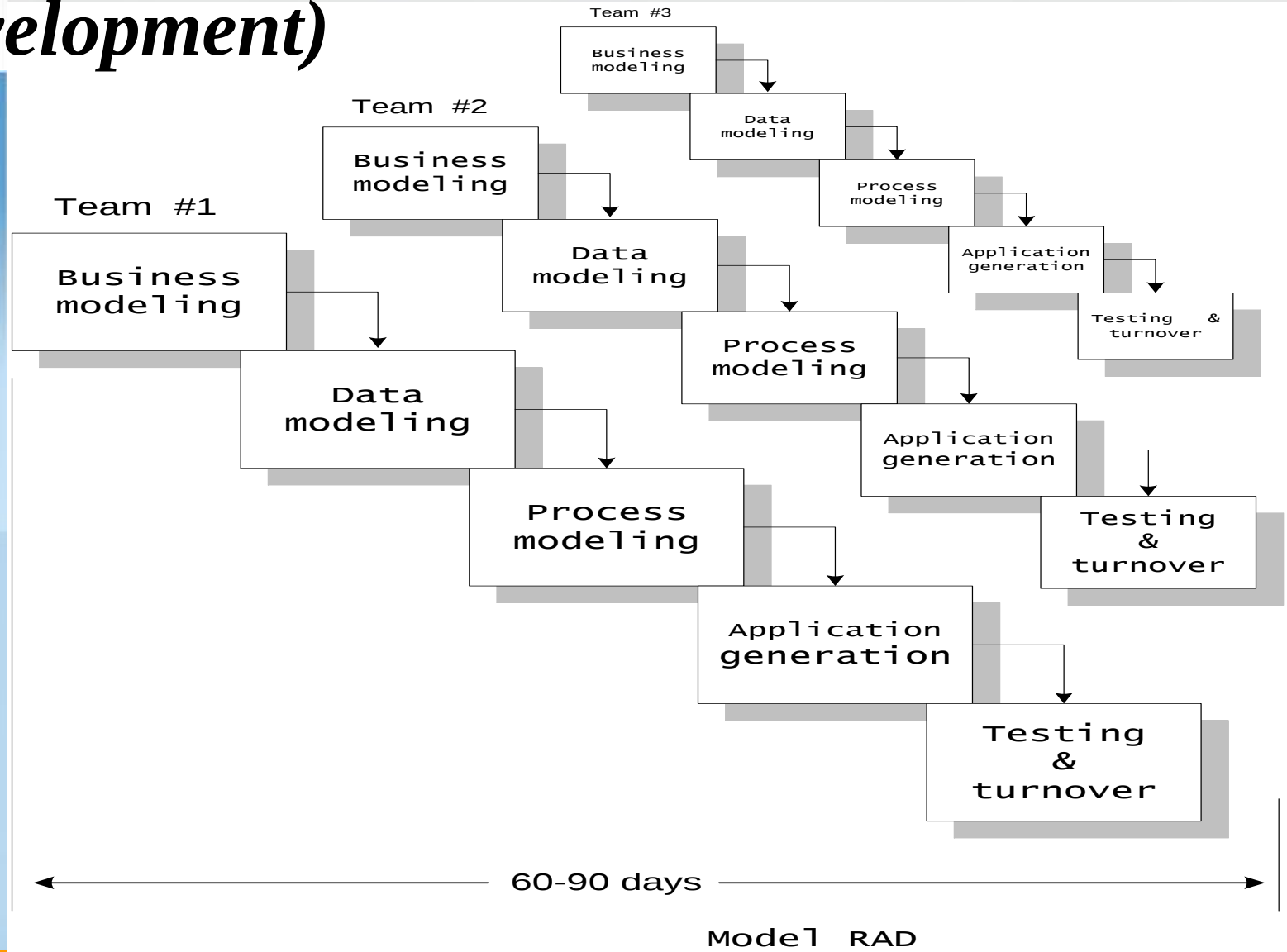


### ***3. MODEL RAD (Rapid Application Development)***

Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.



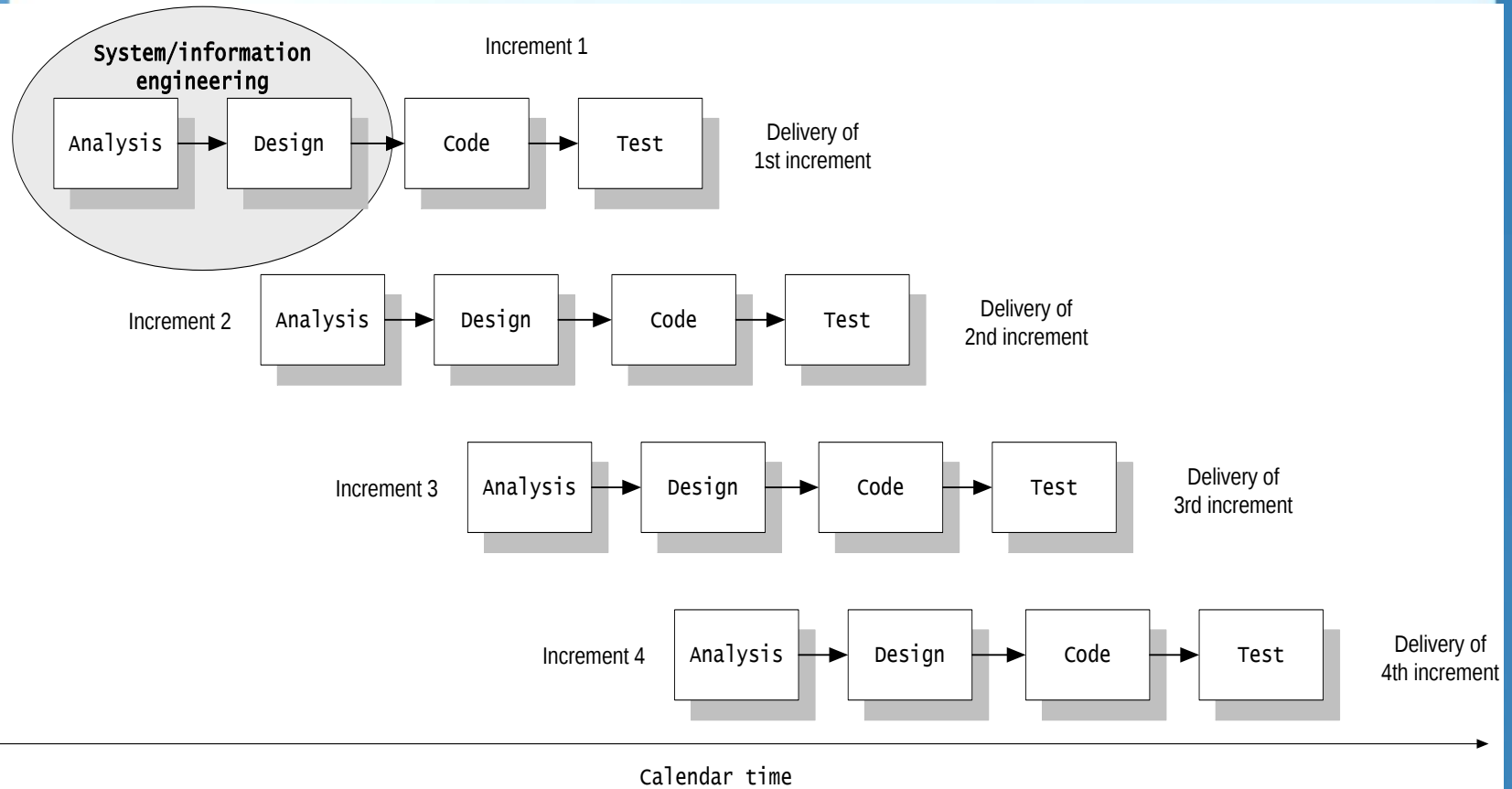
# 3. MODEL RAD (Rapid Application Development)



## ***4. MODEL PROSES PERANGKAT LUNAK EVOLUSIONER***

Model evolusioner ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat lunak yang lebih lengkap sedikit demi sedikit.

# *Model Pertambahan (The Incremental Model)*



Model Pertambahan



# *Model Spiral (The Spiral Model)*

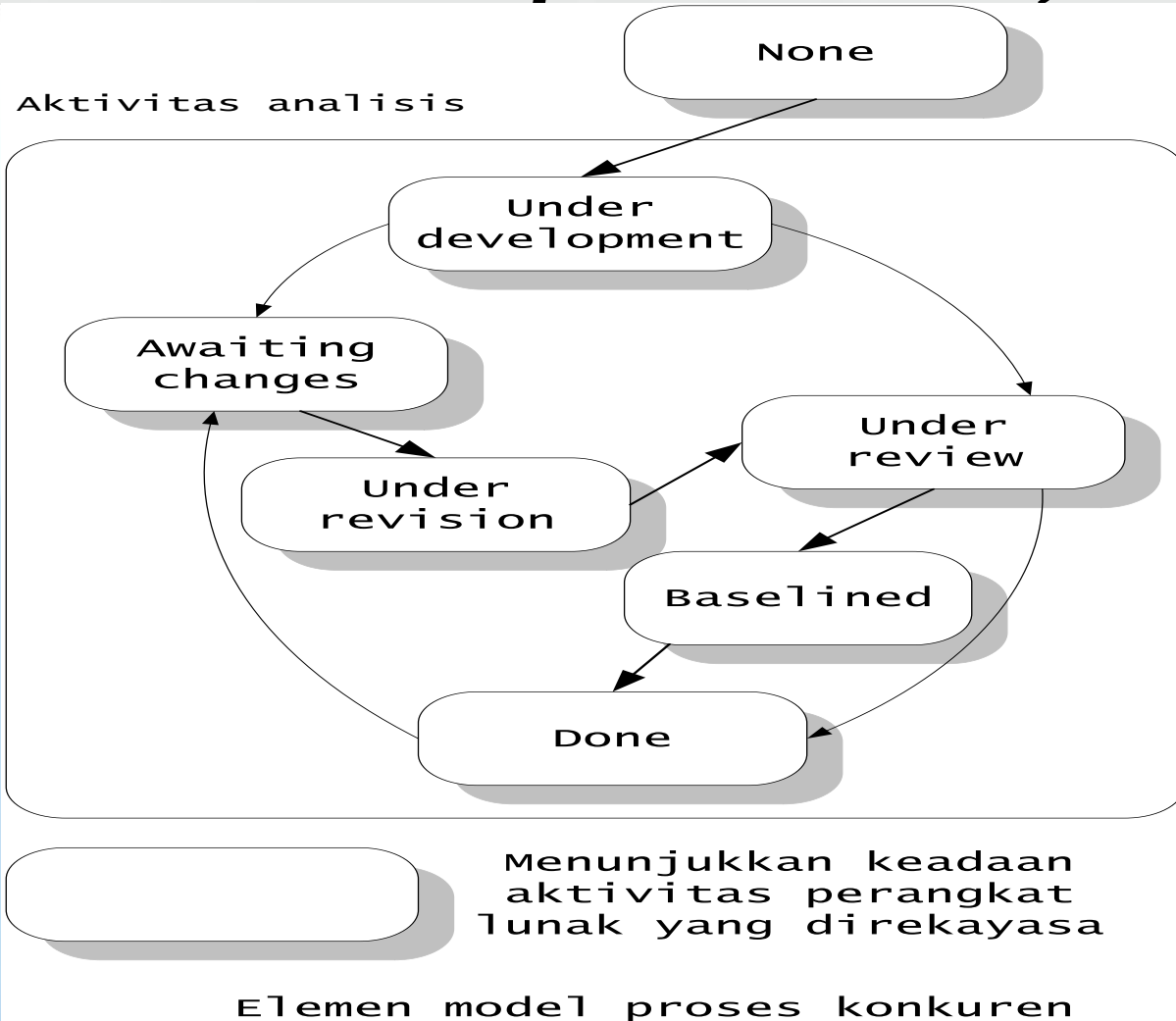
Model spiral merupakan model proses perangkat lunak evolusioner yang merangkai sifat iteratif dari prototype dengan cara control dan aspek sistematis dari model sekuensial linier.

Di dalam model spiral, perangkat lunak dikembangkan di dalam suatu deretan pertambahan. Selama awal iterasi, rilis incremental bisa merupakan sebuah model atau prototipe kertas.

Selama iterasi berikutnya, sedikit demi sedikit dihasilkan versi system rekayasa yang lebih lengkap.



# *Model Perkembangan Konkuren* *(The Concurrent Development Model)*



# MODEL FORMAL

Model formal memungkinkan perekraya perangkat lunak untuk mengkhususkan, mengembangkan, dan memverifikasi system berbasis computer dengan menggunakan notasi matematis yang tepat. Variasi di dalam pendekatan ini disebut juga *clean-room software engineering*, sedang diaplikasikan oleh banyak organisasi pengembang perangkat lunak.



# *Selesai*

