



PENGANTAR REKAYASA PERANGKAT LUNAK

Ketut Agus Seputra, S.ST., M.T



Undiksha

PENDAHULUAN

1950 - 1965

START

- Orientasi batch
- Distribusi terbatas

1975 - 1989

ERA KETIGA

- Sistem terdistribusi
- Embedded Intelligence
- Hardware biaya rendah

ERA KEDUA

- Multiuser
- Real time
- Database

1965 - 1975

ERA KEEMPAT

(Object Oriented
Sistem pakar
Jaringan syaraf tiruan
Komputasi Paralel
Komputasi Jaringan)

1989

RAKAYASA PERANGKAT LUNAK

**STEPHEN
R.SCHACH**

Sebuah disiplin dimana dalam menghasilkan perangkat lunak bebas dari kesalahan dan dalam pengiriman anggaran tepat waktu serta memuaskan keinginan pemakai

FRITZ BAUER

penetapan dan penggunaan prinsip rekayasa dalam rangka memperoleh perangkat lunak yang dapat dipercaya dan dapat bekerja secara efisien pada mesin nyata

IEEE 610.12

sebuah studi pendekatan dan aplikasi secara sistematis, disiplin pengembangan operasi dan pemeliharaan PL yang kesemuanya itu merupakan aplikasi rekayasa yang berkaitan dengan PL



PERANGKAT LUNAK

Karakteristik Perangkat Lunak

Software is developed or engineered, it is not manufactured in the classical sense

- Perangkat lunak adalah suatu produk yang lebih menekankan pada kegiatan rekayasa (engineering) dibandingkan kegiatan manufacturing (rancang bangun di pabrik)

Software doesn't "wear out"

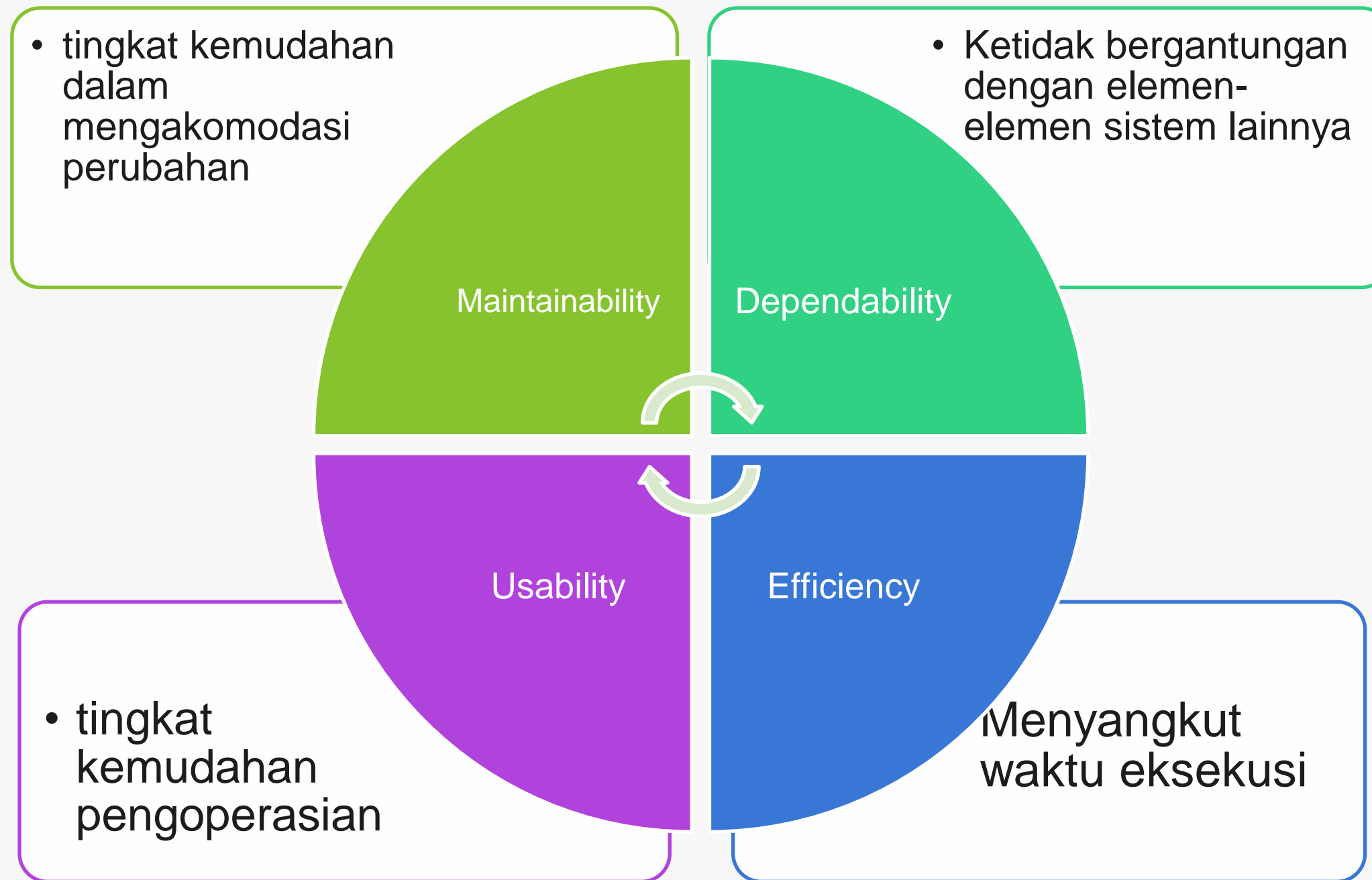
- Perangkat lunak tidak rentan terhadap pengaruh lingkungan yang merusak yang mengakibatkan perangkat keras menjadi usang

most software continues to be custom built

- Sebagian besar software masih dibangun secara custom.
- Kini paradigma baru mulai dikembangkan, yaitu konsep reuseability

PERANGKAT LUNAK

Karakteristik Perangkat Lunak Berkualitas





PERANGKAT LUNAK

Jenis Perangkat Lunak

System software

Application Software

Engineering / scientific software

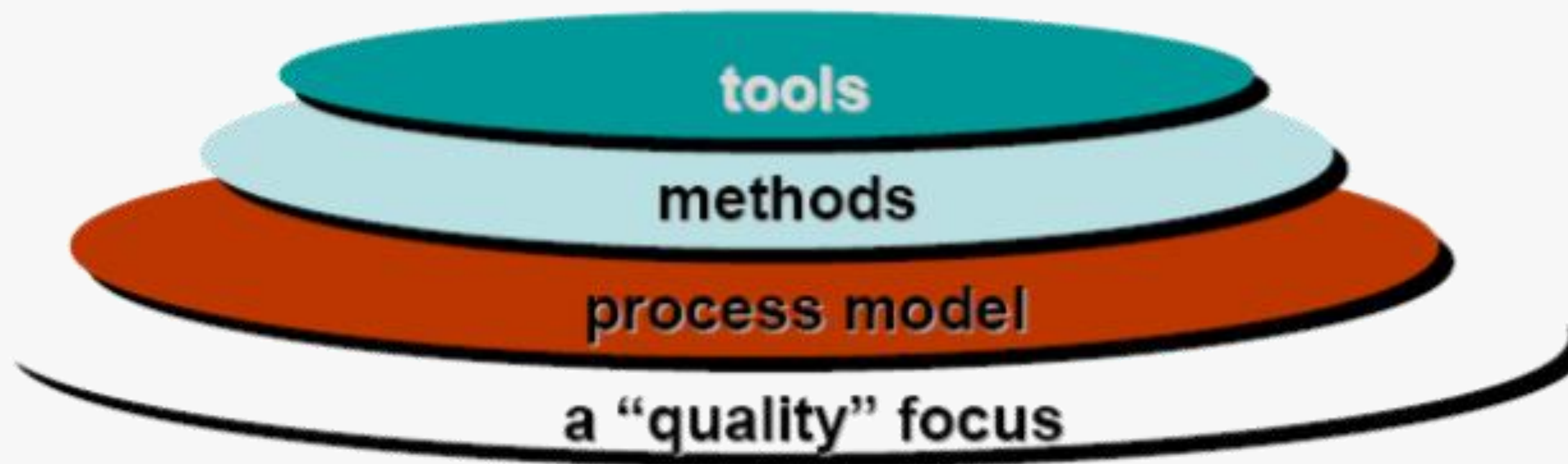
Embedded software

Product line software

Aplikasi web

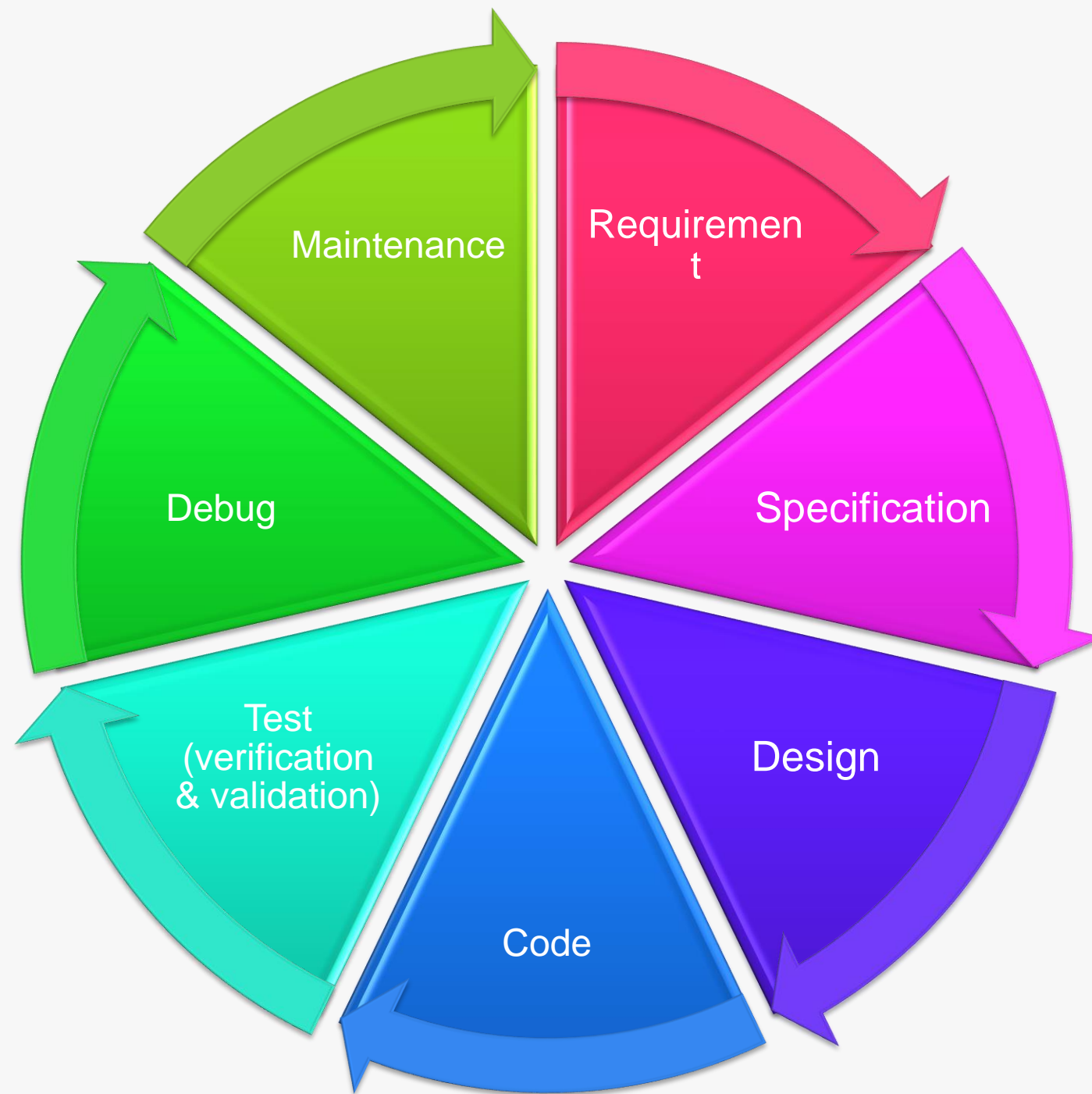
Artificial Intelligence

TEKNOLOGI BERLAPIS PENGEMBANGAN PERANGKAT LUNAK



Lapisan-lapisan Rekayasa Perangkat Lunak

SOFTWARE PROCESS

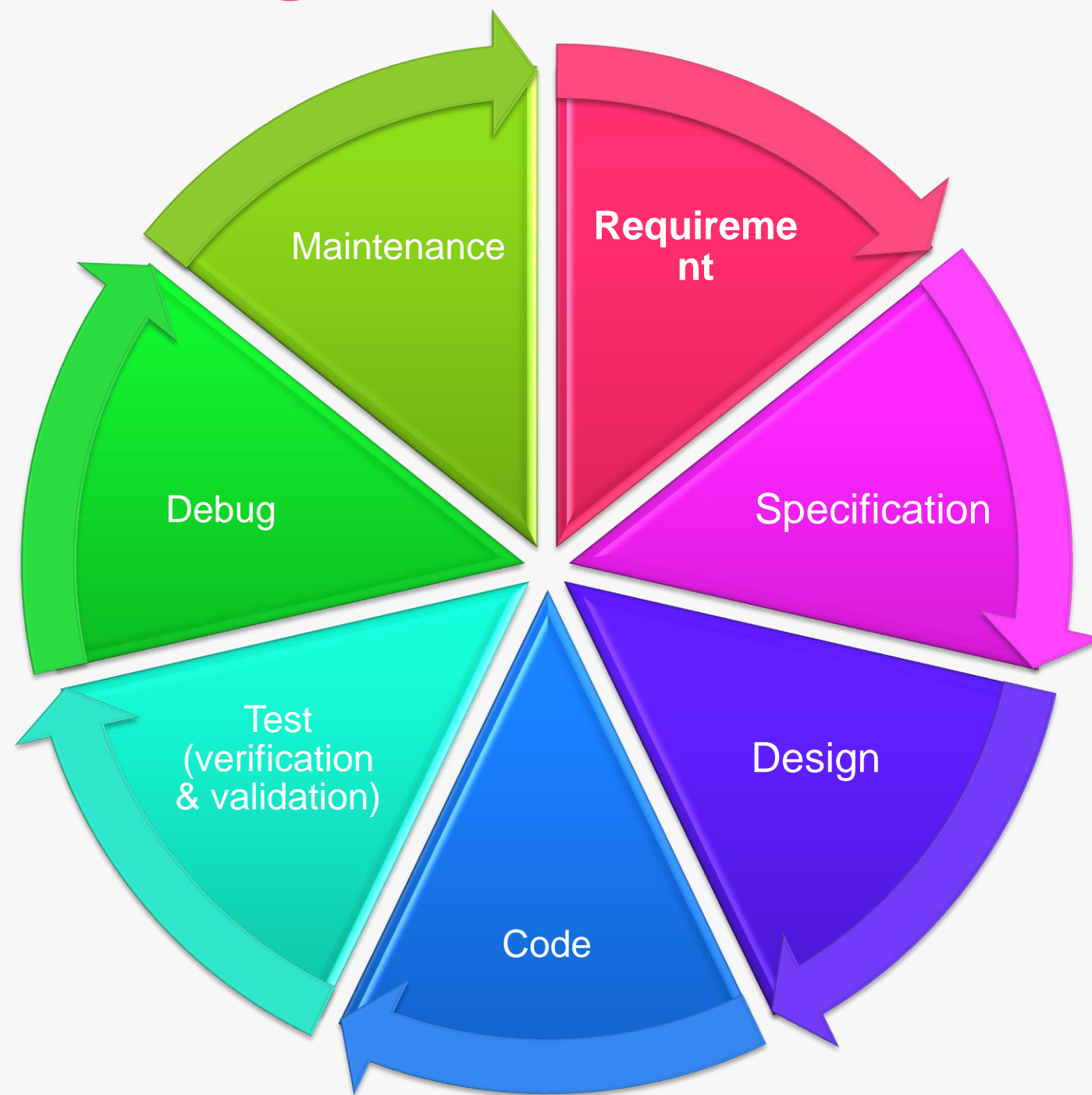


Software Development Life Cycle

Fungsi :

- Panduan kerja bagi para software engineer dengan memetakan peran dan tanggung jawab
- Urutan langkah yang diperlukan dalam pengembangan atau pemeliharaan software
- Kerangka teknis dan manajemen untuk menerapkan metode, alat bantu(tool) serta komponen SDM dalam pengerjaan software

SOFTWARE PROCESS

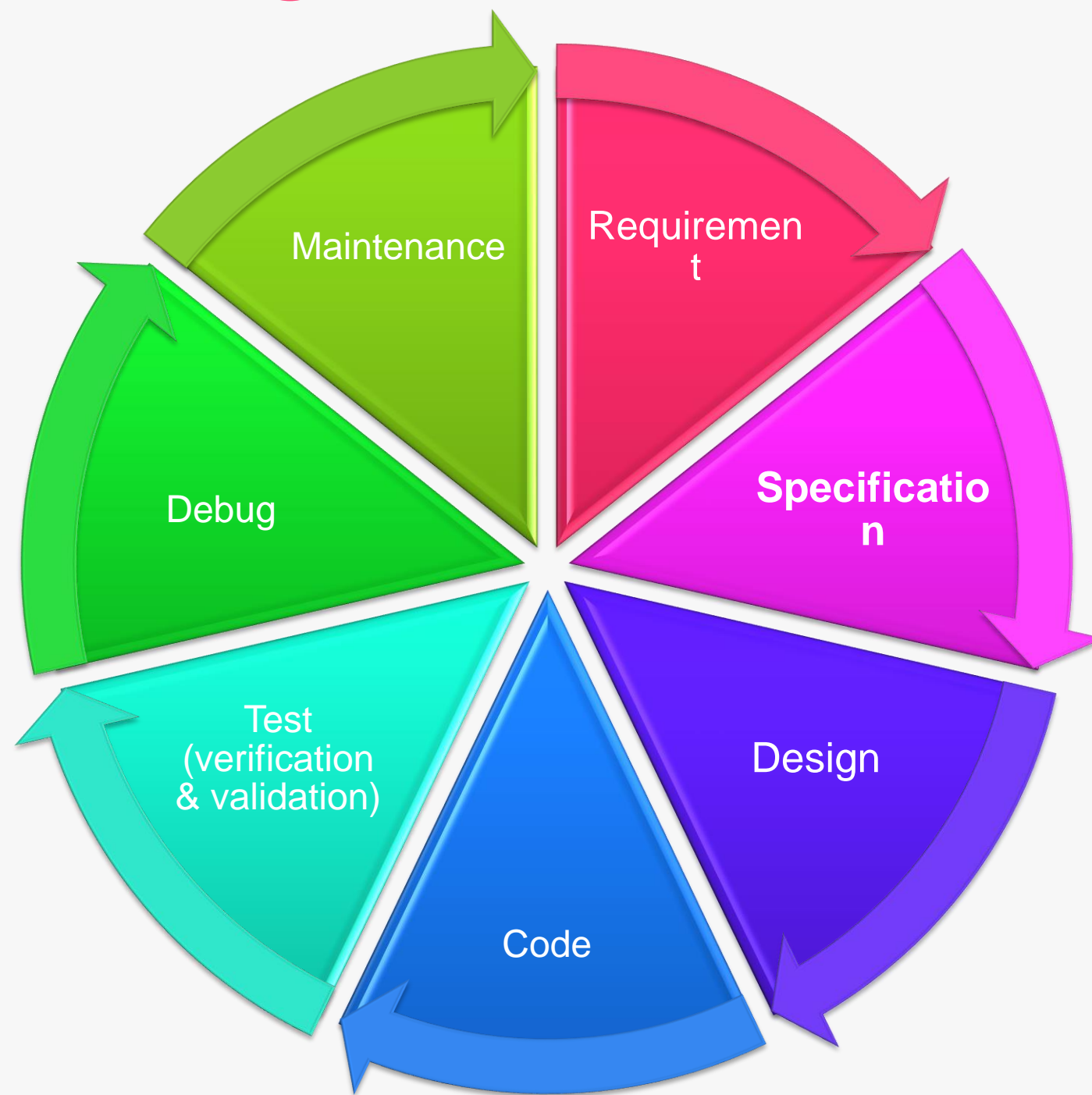


Software Development Life Cycle

Requirement

- ✓ Definisi mengenai perangkat yang dibangun
- ✓ Penetapan kebutuhan pelanggan
- ✓ Sehingga diperoleh simpulan kegunaan system, serta ketersediaan sumber daya waktu, biaya, tenaga
- ✓ Termasuk analisis etika system informasi
 - ✓ Analisis Permasalahan
 - ✓ Analisis Operasional
 - ✓ Analisis Kebutuhan data
 - ✓ Analisis Kapasitas
 - ✓ Analisis Performa
 - ✓ Analisis Keandalan

SOFTWARE PROCESS

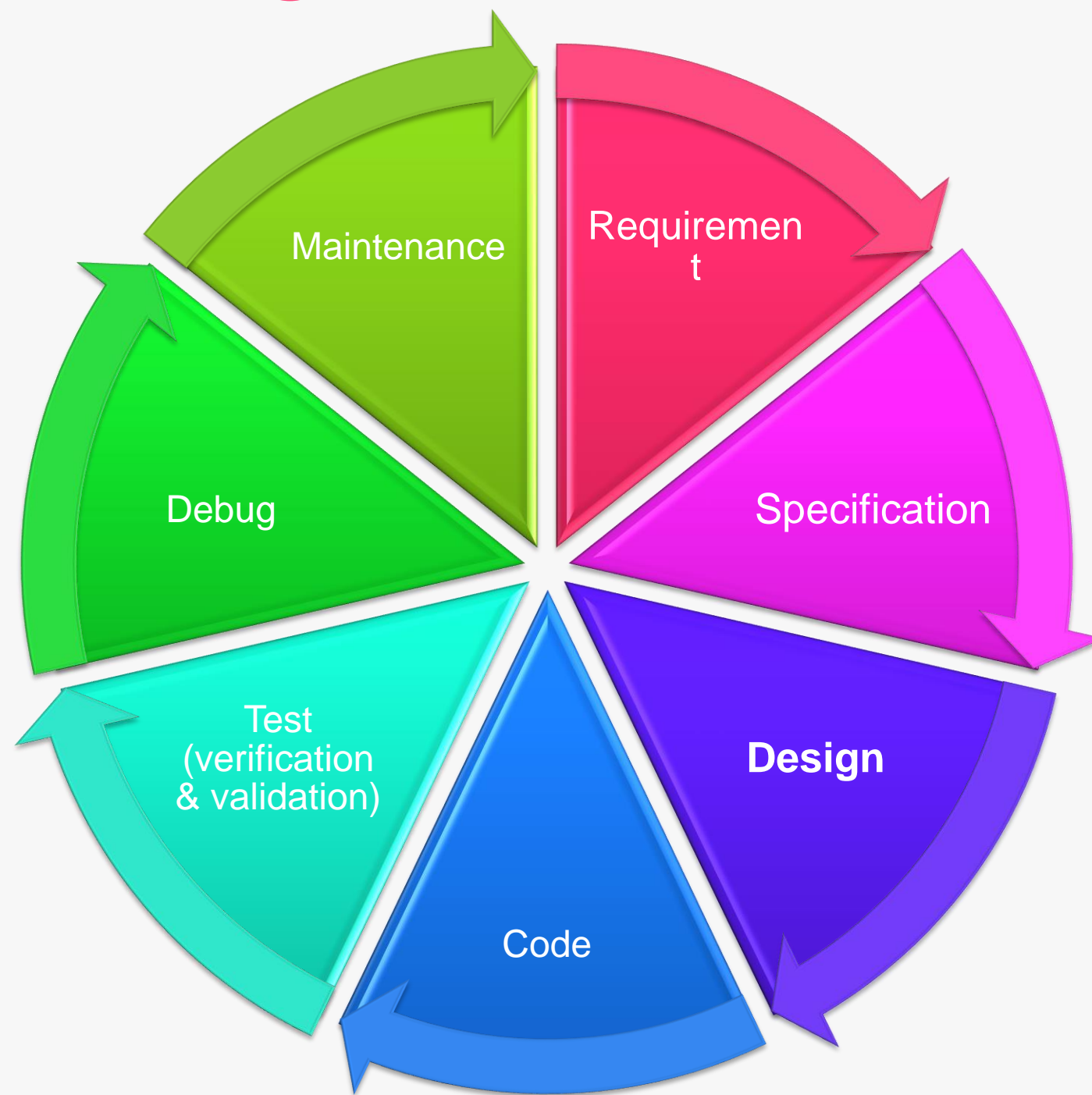


Software Development Life Cycle

Specification

- ✓ Kebutuhan pelanggan (requirement) yang telah ditetapkan ditransformasikan ke dalam kebutuhan system
- ✓ Diperoleh spesifikasi detail mengenai produk perangkat lunak yang akan dibangun antara lain seputar fungsionalitasnya, kebutuhan perangkat keras dan perangkat lunak pendukung pengembangan

SOFTWARE PROCESS



Software Development Life Cycle

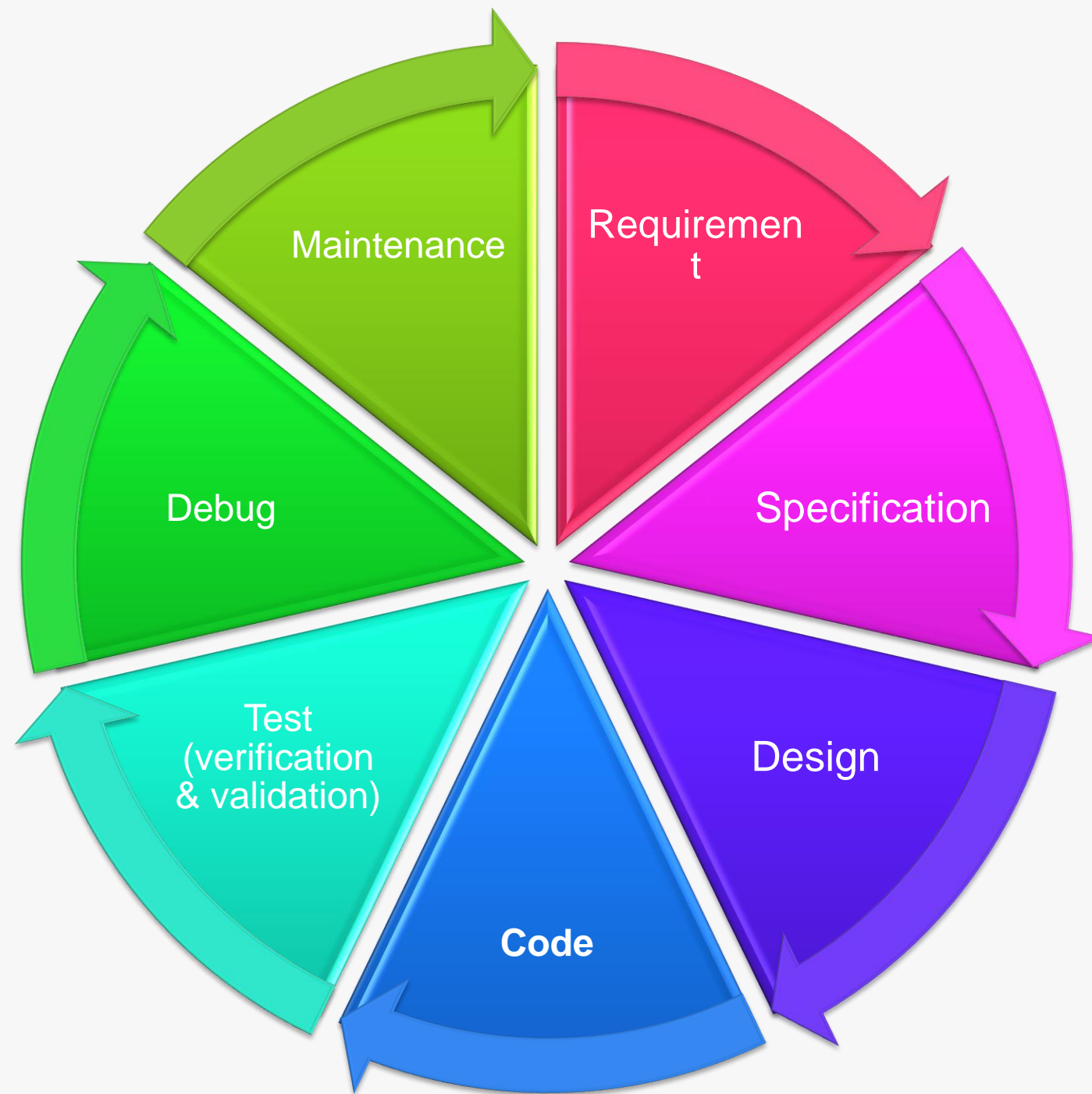
Design

- ✓ Hasil analisis kebutuhan dan spesifikasi sistem dibentuk dalam suatu model
- ✓ harus mendefinisikan bagaimana data dikonstruksikan
- ✓ bagaimana fungsi-fungsi diimplementasikan sebagai sebuah arsitektur perangkat lunak
- ✓ bagaimana detail prosedur akan diimplementasikan
- ✓ bagaimana interface ditandai
- ✓ diperoleh penggambaran sistem dalam bentuk model semacam use case diagram, data flow diagram, sequence diagram, entity relationship diagram, dan lain-lain.

SOFTWARE PROCESS

Code

- ✓ Hasil rancangan (model) dari tahapan sebelumnya diterjemahkan dalam bentuk coding program pada sebuah bahasa pemrograman

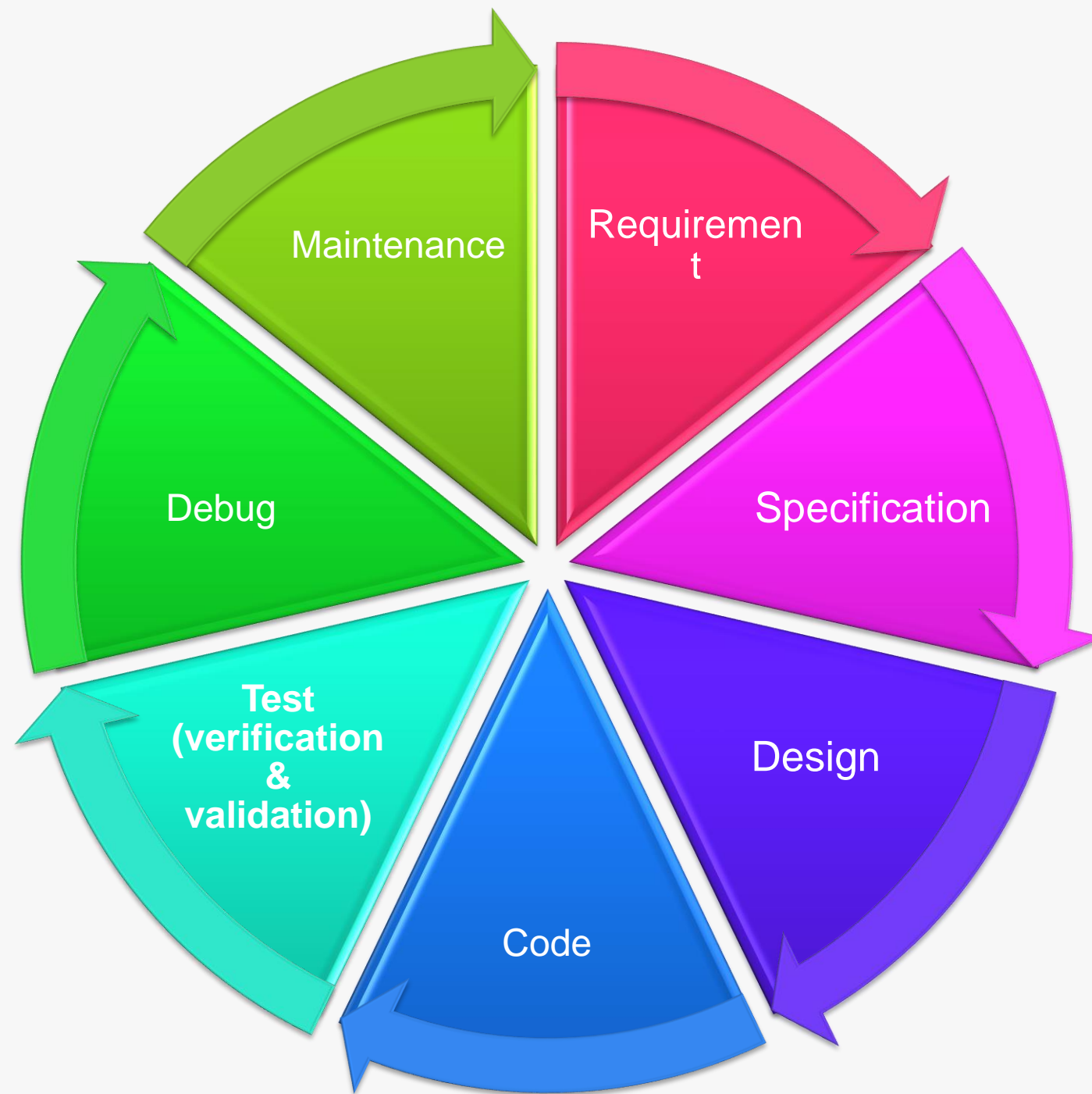


Software Development Life Cycle

SOFTWARE PROCESS

Test (verification & validation)

- ✓ Proses pengujian untuk memastikan apakah aplikasi yang dibangun sudah sesuai dengan spesifikasi dan kebutuhan pelanggan yang ditetapkan awal ditetapkan

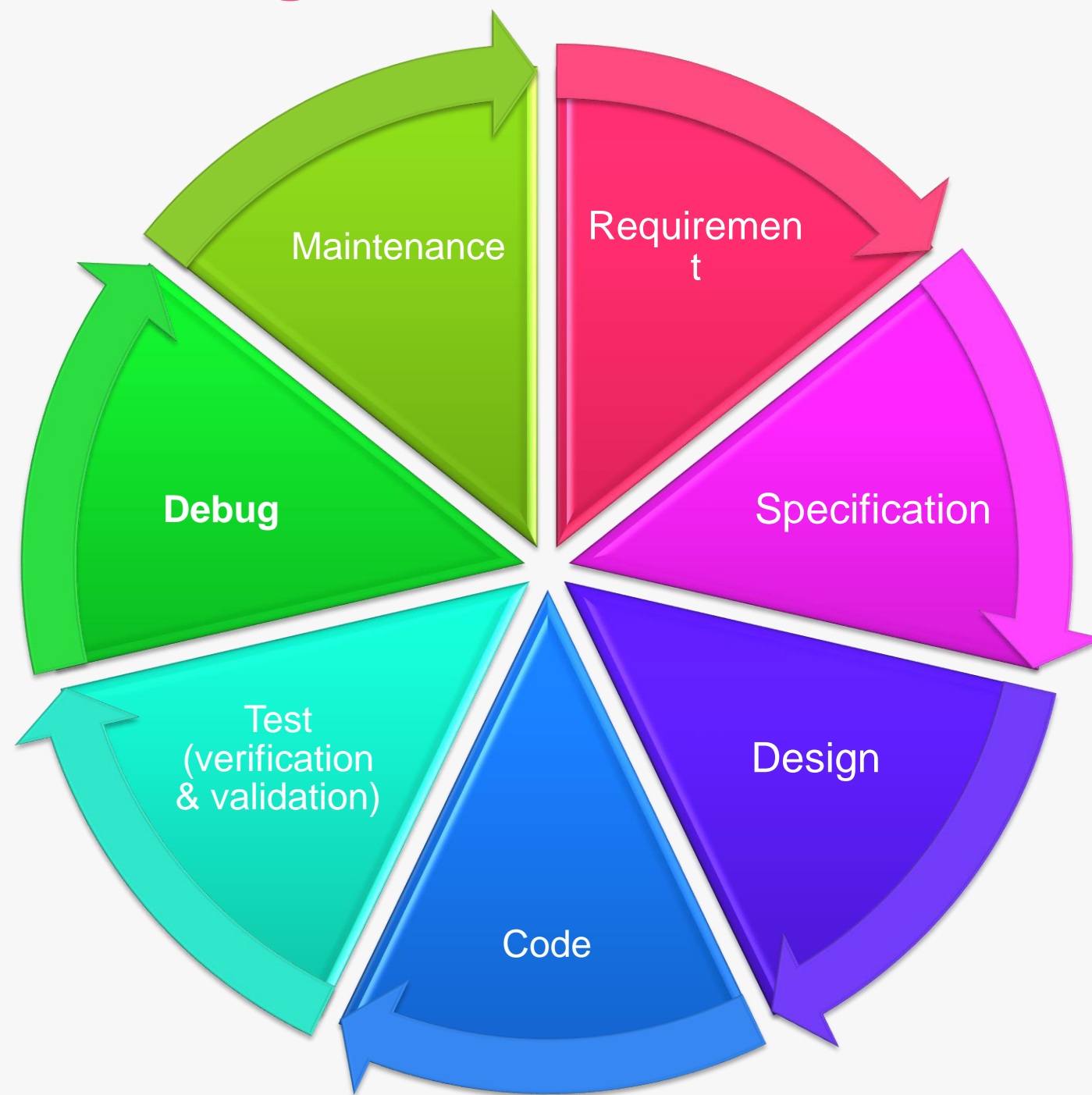


Software Development Life Cycle

SOFTWARE PROCESS

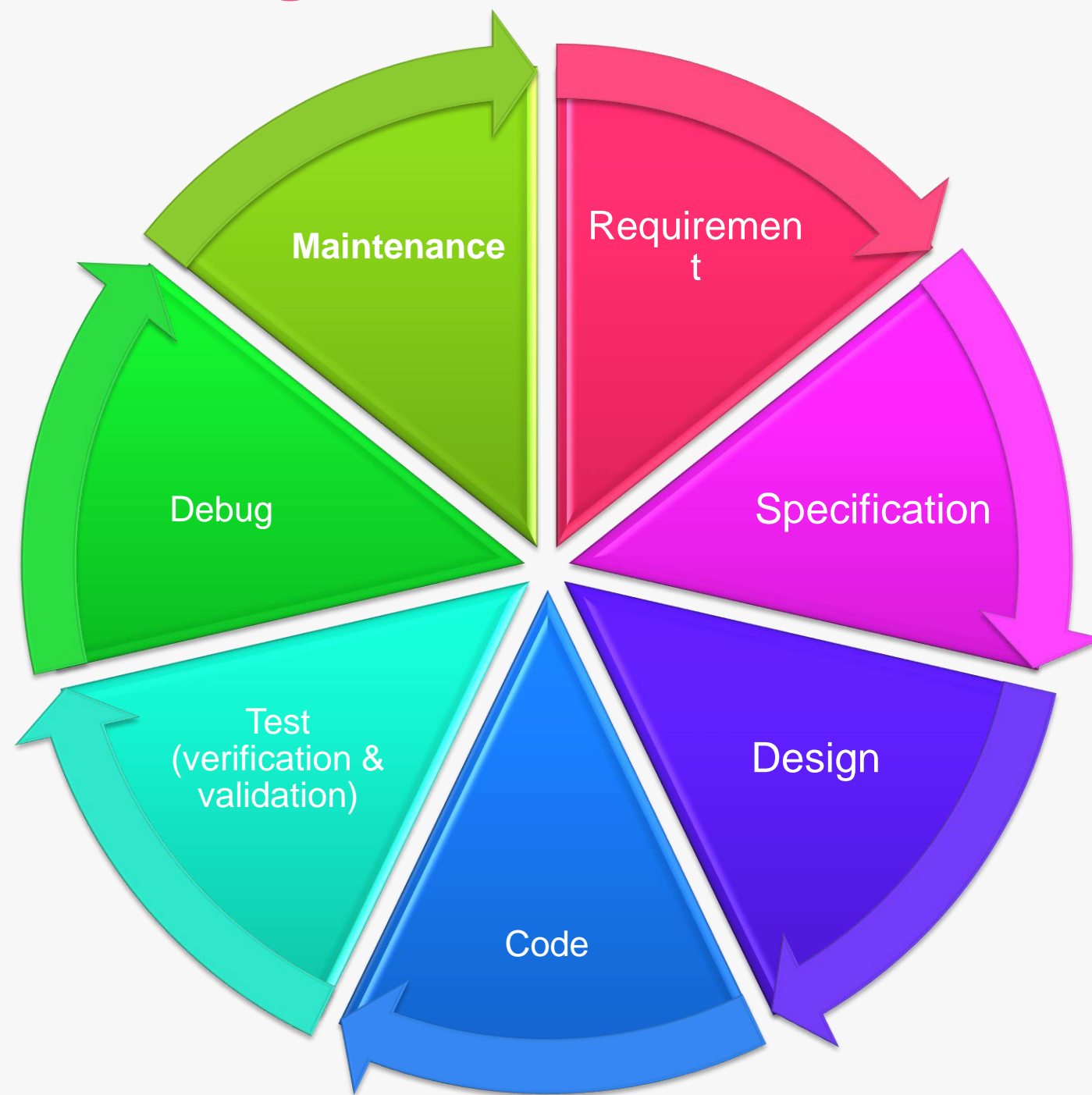
Debug

- ✓ Proses perbaikan yang diperlukan apabila pada fase pengujian masih ditemukan adanya kesalahan



Software Development Life Cycle

SOFTWARE PROCESS



Software Development Life Cycle

Maintenance

- ✓ berfokus pada perubahan (change), yang dihubungkan dengan koreksi kesalahan
- ✓ penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang
- ✓ perubahan sehubungan dengan perkembangan yang disebabkan oleh perubahan kebutuhan pelanggan
 - Koreksi (corrective)
 - Adaptasi (adaptive)
 - Pengembangan (perfective)
 - Pencegahan (preventive)

SOFTWARE PROCESS

Bagaimana menggunakan Software Process Yang Baik ?

Understandability

- Proses secara eksplisit didefinisikan

Visibility

- Aktivitas proses memberi hasil yang jelas

Supportability

- Proses dapat didukung oleh teknologi semacam CASE tools

Acceptability

- Penerimaan atas proses yang terdefinisi oleh engineer

Reliability

- Proses didisain dengan suatu metode untuk menghindar dari kesalahan

Robustness

- Proses dapat terus dilanjutkan meskipun terdapat masalah

Maintainability

- Proses dapat mengadaptasi terhadap permintaan perubahan ataupun perbaikan

Rapidity

- Proses dapat diselesaikan dalam waktu yang relatif cepat



SOFTWARE PROCESS



PENTING aktifitas pelindung (umbrella activities) ?

Kegiatan-kegiatan khusus di dalam kategori ini menyangkut :

- ✓ Manajemen proyek PL : melindungi agar PL yang ada hasilnya bagus
- ✓ Formal technical review, contoh : menemui user dan mengecek kebutuhannya untuk analisis
- ✓ Software quality assurance (jaminan kualitas PL)
- ✓ Manajemen konfigurasi PL
- ✓ Pembuatan dan penyiapan dokumen : sebagai senjata jika user tiba-tiba meminta tambahan fungsi PL
- ✓ Reusability management (manajemen reusabilitas)
- Measurement (pengukuran)
- ✓ Risk Management (manajemen resiko)



TAHAP DESIGN | PERANCANGAN SISTEM

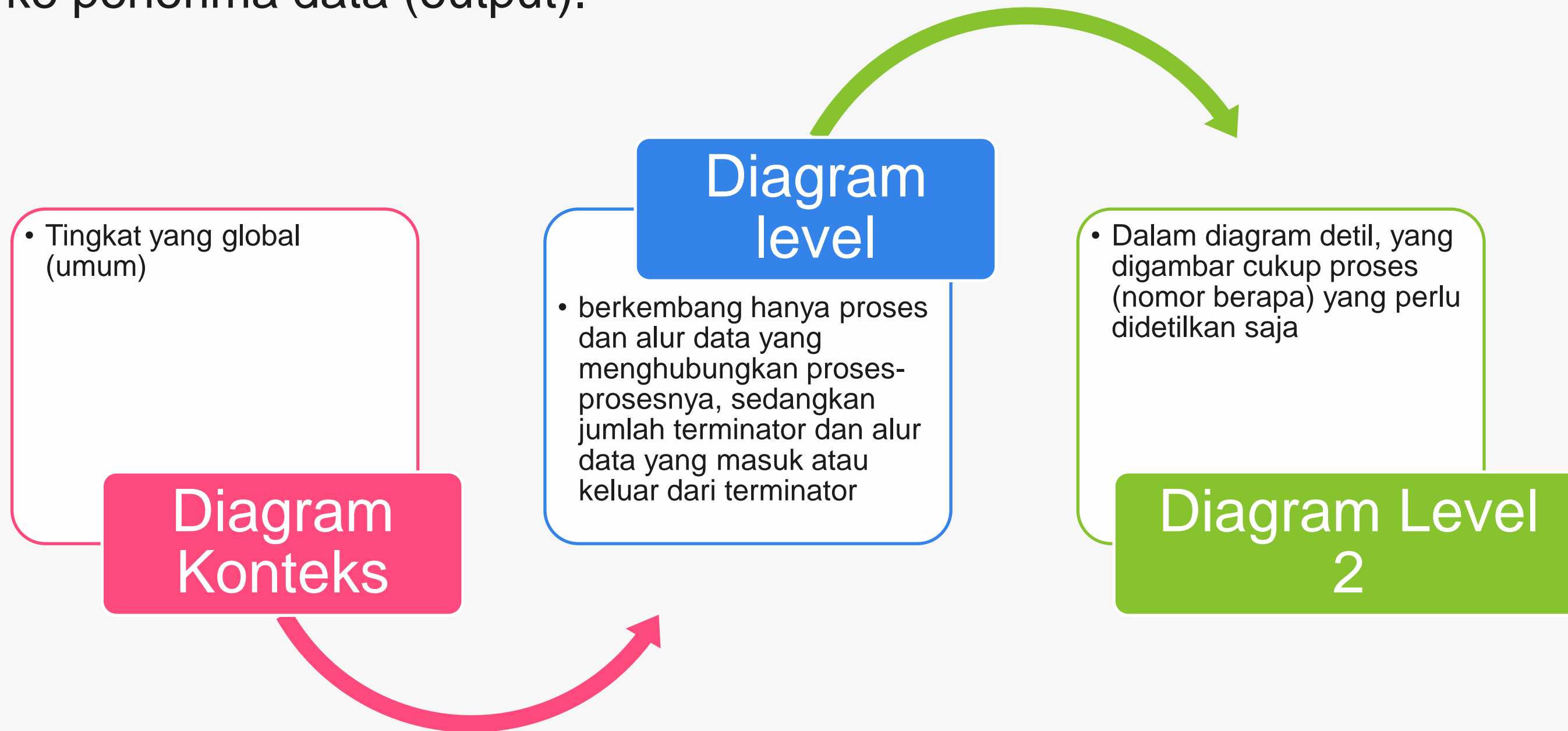
KONSEP PERANCANGAN TERSTRUKTUR.

Beberapa rancangan yang disiapkan:

- ✓ DATA FLOW DIAGRAM (DFD)
- ✓ Flowchart
- ✓ Entity Relationship Diagram (ERD) Perancangan Interface

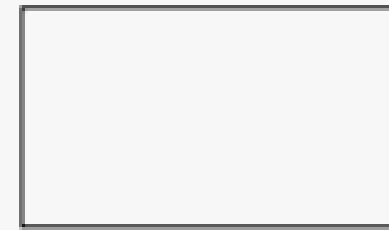
DATA FLOW DIAGRAM (DFD)

DFD menggambarkan aliran data dari sumber pemberi data (input) ke penerima data (output).

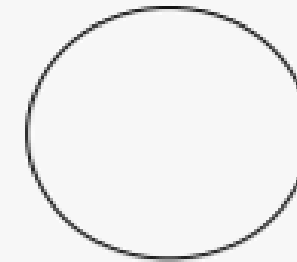


DATA FLOW DIAGRAM (DFD)

KOMPONEN



Terminator



Proses



Data Store



Alur Data

- **Terminator** mewakili entitas eksternal yang berkomunikasi dengan sistem yang sedang dikembangkan.
- Pemberian nama **proses** dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek), seperti **Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS**.
- Pemberian nama **proses** dilakukan dengan menggunakan kata kerja transitif (kata kerja yang membutuhkan obyek), seperti Menghitung Gaji, Mencetak KRS, Menghitung Jumlah SKS.
- Suatu **data store** dihubungkan dengan alur data hanya pada komponen proses

DATA FLOW DIAGRAM (DFD)

DIAGRAM KONTEKS

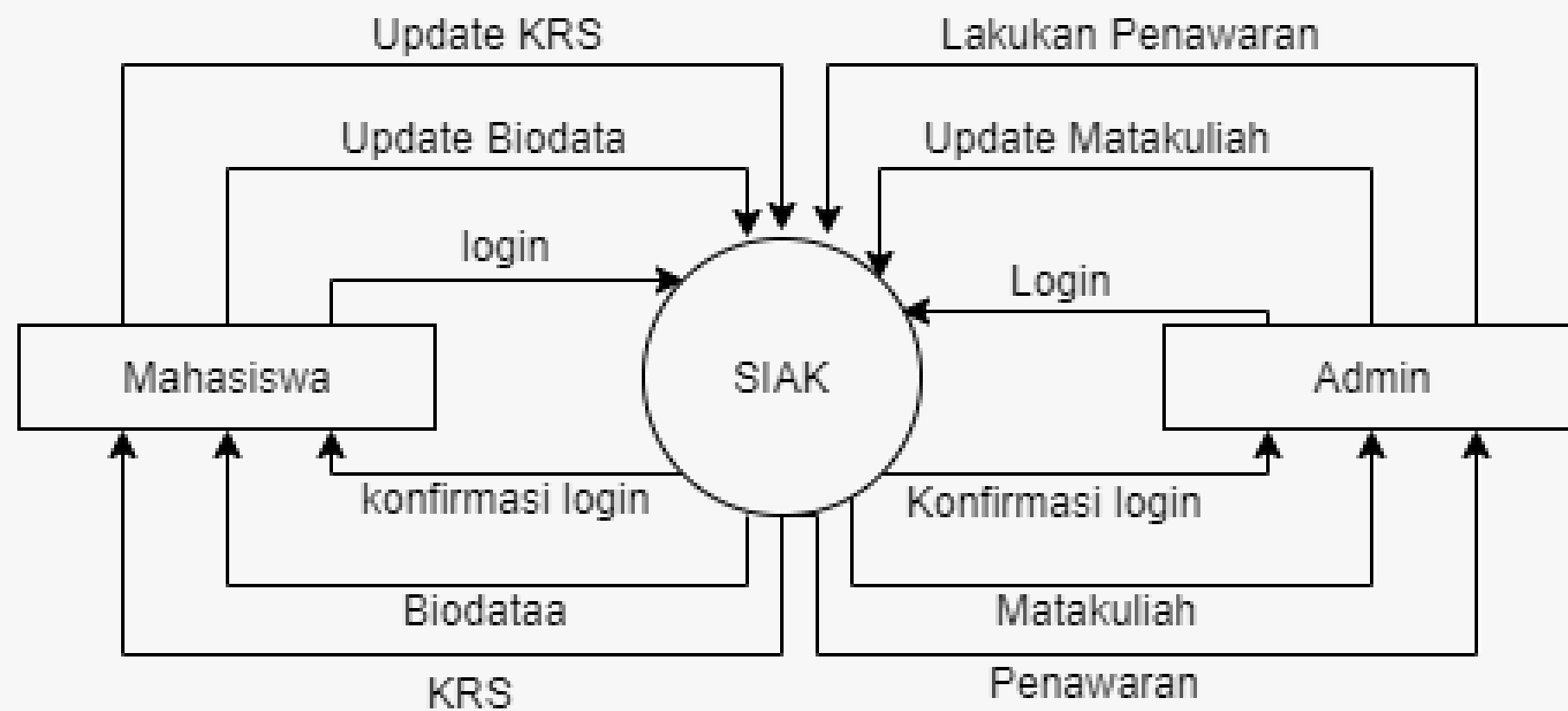
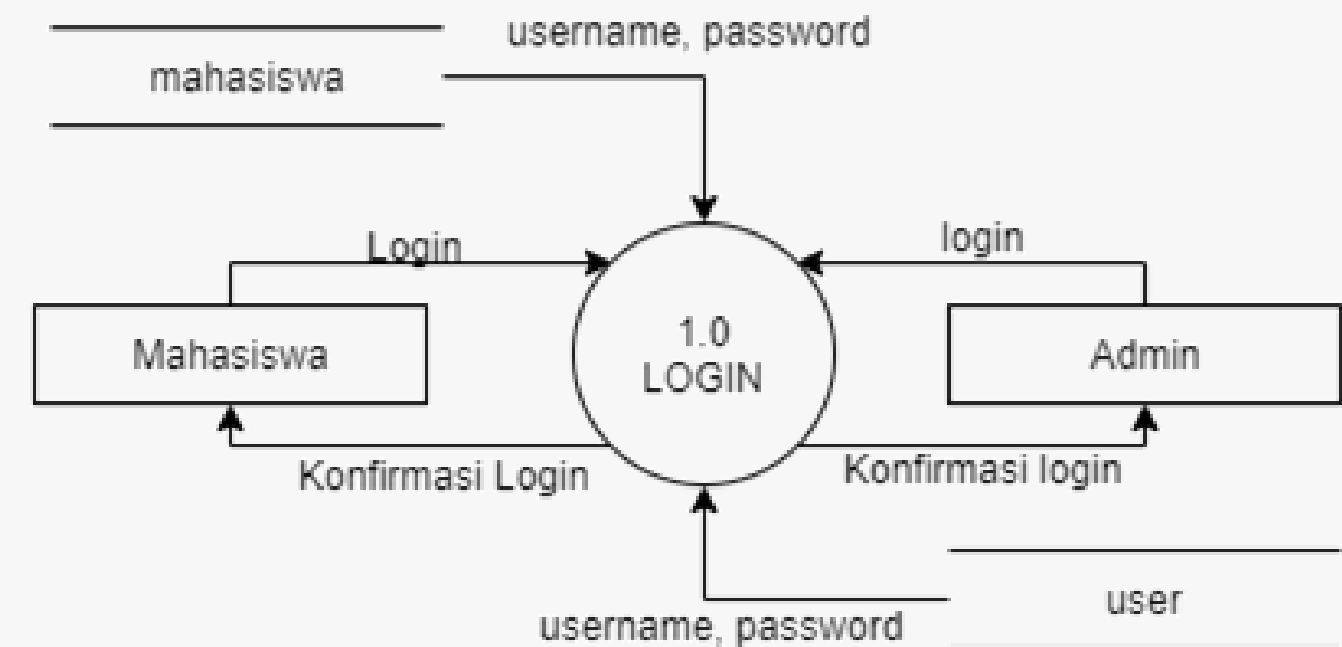


DIAGRAM Level 0



TERIMA KASIH

