Nama : Rizal prasetyo

Kelas : 1IKRA

Nim : 250202965

Perbedaan Monolithic Kernel, Microkernel, dan Layered Architecture

**1. Monolithic Kernel**

Monolithic kernel adalah jenis arsitektur sistem operasi di mana seluruh komponen utama system seperti manajemen proses, manajemen memori, sistem file, dan driver perangkat keras berjalan di dalam satu ruang kernel kernel space.  
Semua komponen saling terhubung langsung dan dapat saling memanggil fungsi satu sama lain tanpa pembatas.

Pada model ini, kernel berperan sebagai inti tunggal yang menangani semua tugas penting sistem operasi. Karena semua komponen berada di ruang kernel, komunikasi antarbagian sistem berlangsung sangat cepat. Ketika program pengguna melakukan pemanggilan sistem system call, kernel akan menanganinya secara langsung tanpa melalui perantara tambahan.

Kelebihan utama dari arsitektur ini adalah kinerja yang tinggi, karena semua komponen bekerja dalam satu lingkungan yang sama tanpa biaya tambahan untuk komunikasi antar proses. Namun, kelemahan besarnya adalah keamanan dan stabilitas yang rendah. Jika satu bagian kernel, seperti driver perangkat, mengalami kerusakan atau kesalahan, seluruh sistem dapat mengalami crash. Selain itu, karena seluruh komponen bergabung menjadi satu kode besar, proses pengembangan, debugging, dan perawatan sistem menjadi jauh lebih rumit.

Secara konsep, monolithic kernel menawarkan efisiensi tinggi, tetapi mengorbankan keamanan dan fleksibilitas.

**2. Microkernel**

Microkernel adalah kebalikan dari monolithic kernel. Dalam arsitektur ini, hanya komponen-komponen paling dasar yang dijalankan di ruang kernel.  
Fungsi inti kernel terbatas pada hal-hal seperti komunikasi antar proses inter-process communication, manajemen memori dasar, penjadwalan CPU, dan penanganan interupsi.

Komponen lain seperti sistem file, manajemen perangkat keras, jaringan, dan antarmuka sistem ditempatkan di luar kernel, yaitu di ruang pengguna user space.  
Setiap layanan ini dijalankan sebagai proses terpisah dan berinteraksi satu sama lain melalui mekanisme pengiriman pesan message passing.

Dengan pemisahan ini, setiap layanan berjalan secara independen. Jika salah satu layanan mengalami kegagalan, bagian lain dari sistem tidak akan ikut terpengaruh. Hal ini menjadikan microkernel jauh lebih stabil dan aman dibandingkan model monolithic.  
Selain itu, model ini sangat modular: pengembang dapat memperbarui, mengganti, atau menghapus suatu layanan tanpa harus memodifikasi seluruh kernel.

Namun, ada konsekuensi terhadap performa. Karena komunikasi antar layanan dilakukan melalui mekanisme pesan dan melibatkan perpindahan konteks antara ruang pengguna dan ruang kernel, microkernel biasanya memiliki kecepatan yang sedikit lebih rendah dibandingkan monolithic kernel.  
Selain itu, implementasinya lebih kompleks karena membutuhkan pengaturan komunikasi antar proses yang efisien.

Dengan kata lain, microkernel menekankan **keamanan, modularitas, dan stabilitas**, tetapi dengan pengorbanan pada **kecepatan dan efisiensi eksekusi**.

**3. Layered Architecture**

Layered architecture atau arsitektur berlapis merupakan pendekatan desain di mana sistem operasi dibagi ke dalam beberapa lapisan atau tingkat layer yang tersusun secara hierarkis.  
Setiap lapisan memiliki fungsi dan tanggung jawab tertentu, serta hanya dapat berinteraksi dengan lapisan yang berada tepat di atas atau di bawahnya.

Lapisan paling bawah biasanya berhubungan langsung dengan perangkat keras, sedangkan lapisan paling atas berinteraksi dengan pengguna atau aplikasi.  
Misalnya, satu lapisan bertanggung jawab untuk manajemen memori, lapisan di atasnya mengatur sistem file, dan lapisan berikutnya menangani interaksi pengguna.

Tujuan utama dari pendekatan ini adalah menciptakan sistem yang **terstruktur, terorganisasi, dan mudah dipelihara**.  
Dengan pembagian yang jelas antara fungsi setiap lapisan, pengembangan dan pengujian dapat dilakukan secara terpisah. Jika ada kesalahan dalam satu lapisan, pengembang dapat memperbaikinya tanpa memengaruhi bagian lain dari sistem.

Kelebihan utama dari layered architecture adalah **kemudahan pemeliharaan, modularitas, dan kejelasan desain**. Sistem menjadi lebih mudah dipahami dan dikembangkan.  
Namun, model ini juga memiliki kelemahan, terutama dalam hal efisiensi. Karena setiap permintaan harus melewati beberapa lapisan sebelum mencapai perangkat keras, waktu respon sistem menjadi lebih lama dibandingkan model lain.  
Selain itu, perancangan batas antar lapisan harus dilakukan dengan hati-hati agar tidak terjadi tumpang tindih fungsi atau ketergantungan yang tidak diinginkan.

Dengan demikian, layered architecture menonjol dalam hal **desain yang sistematis dan mudah dikelola**, namun **tidak seefisien** model lain dalam hal kinerja eksekusi.