Nama : Silvia Putri Herlieno

No. BP: 2111082043

Kelas : TRPL 3B

**Arsitektur Sistem Terdistribusi**

Arsitektur adalah Suatu rancangan untuk penyusunan komponen-komponen suatu sistem, dimana rancangan tersebut mengidentifikasi komponen serta fungsi masing-masing komponen, konektifitas antar komponen serta pemetaan fungsionalitas komponen.

Perangkat lunak berbasis cloud, tulang punggung sistem terdistribusi, adalah jaringan server rumit yang dapat diakses oleh siapa saja yang memiliki koneksi internet. Dalam sistem terdistribusi, komponen dan konektor mengatur dirinya sendiri sedemikian rupa sehingga memudahkan komunikasi. Komponen adalah modul dengan antarmuka terdefinisi dengan baik yang dapat diganti atau digunakan kembali. Demikian pula konektor merupakan penghubung komunikasi antar modul yang memediasi koordinasi atau kerjasama antar komponen.

Sistem terdistribusi secara luas dibagi menjadi dua konsep penting - arsitektur perangkat lunak (selanjutnya dibagi menjadi arsitektur berlapis, arsitektur berbasis objek, arsitektur pusat data, dan arsitektur berbasis peristiwa) dan arsitektur sistem (selanjutnya dibagi menjadi arsitektur klien-server dan peer- arsitektur to-peer).

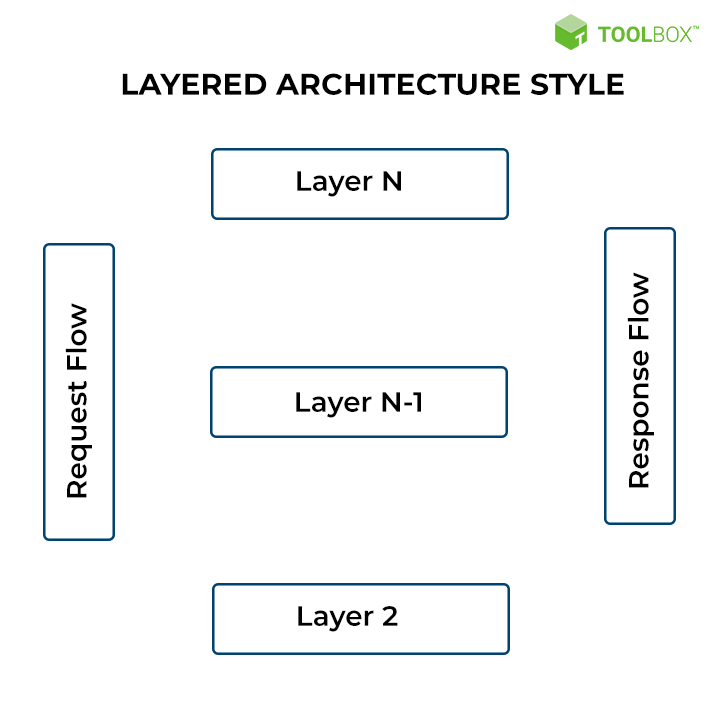
1. Arsitektur perangkat lunak

Arsitektur perangkat lunak adalah organisasi logis dari komponen perangkat lunak dan interaksinya dengan struktur lain. Ini berada pada tingkat yang lebih rendah dari arsitektur sistem dan berfokus sepenuhnya pada komponen; misalnya, web front end dari sistem ecommerce adalah sebuah komponen. Empat gaya arsitektur utama sistem terdistribusi dalam komponen perangkat lunak meliputi:

1. Arsitektur berlapis (Layered architecture)

Arsitektur berlapis memberikan pendekatan modular terhadap perangkat lunak. Dengan memisahkan masing-masing komponen maka lebih efisien. Misalnya, model interkoneksi sistem terbuka (OSI) menggunakan arsitektur berlapis untuk hasil yang lebih baik. Ia melakukan ini dengan menghubungi lapisan secara berurutan, yang memungkinkannya mencapai tujuannya. Dalam beberapa kasus, penerapan arsitektur berlapis berada dalam koordinasi lintas lapisan. Di bawah lapisan silang, interaksi dapat melewati lapisan mana pun yang berdekatan hingga lapisan tersebut memenuhi permintaan dan memberikan hasil kinerja yang lebih baik.

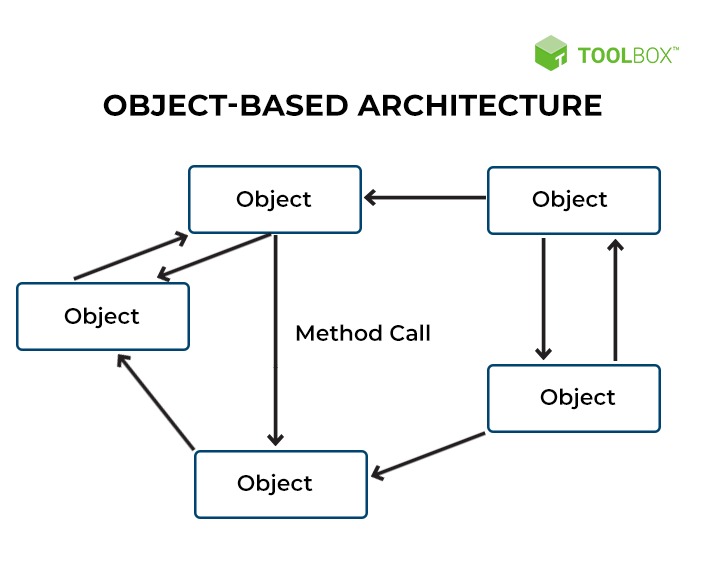
Arsitektur berlapis adalah jenis perangkat lunak yang memisahkan komponen menjadi beberapa unit. Permintaan datang dari atas ke bawah, dan tanggapannya datang dari bawah ke atas. Keuntungan dari arsitektur berlapis adalah menjaga segala sesuatunya tetap teratur dan memodifikasi setiap lapisan secara independen tanpa mempengaruhi sistem lainnya.



1. Arsitektur berbasis objek (Object-based architecture)

Arsitektur berbasis objek berpusat pada susunan objek yang digabungkan secara longgar tanpa arsitektur khusus seperti lapisan. Berbeda dengan arsitektur berlapis, arsitektur berbasis objek tidak harus mengikuti langkah apa pun secara berurutan. Setiap komponen adalah sebuah objek, dan semua objek dapat berinteraksi melalui antarmuka (atau konektor). Dalam arsitektur berbasis objek, interaksi antar komponen dapat terjadi melalui pemanggilan metode langsung.

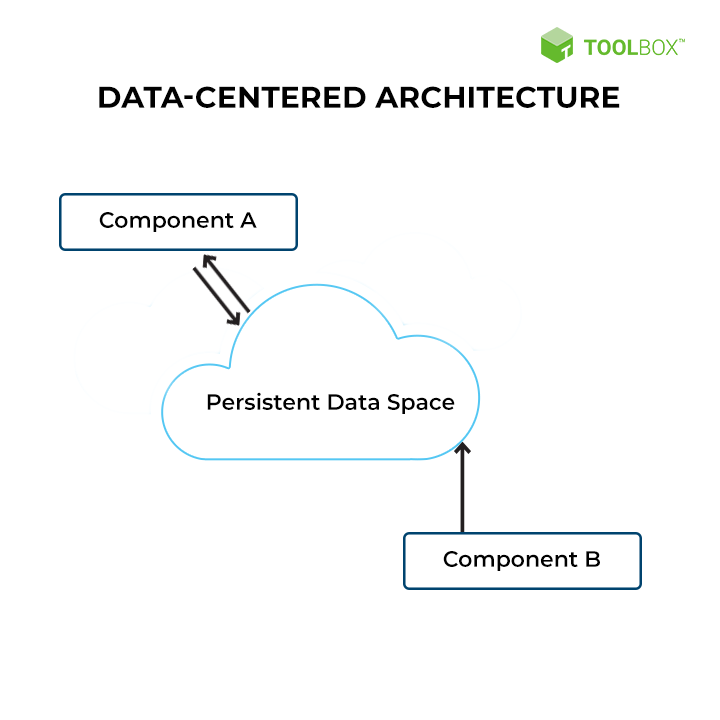
Pada intinya, komunikasi antar objek terjadi melalui pemanggilan metode, sering disebut panggilan prosedur jarak jauh (RPC). Sistem RPC yang populer mencakup Java RMI dan Layanan Web serta Panggilan REST API. Pertimbangan desain utama dari arsitektur ini adalah bahwa arsitektur tersebut kurang terstruktur. Di sini, komponen sama dengan objek, dan konektor sama dengan RPC atau RMI.



1. Arsitektur berpusat pada data (Data-centered architecture)

Arsitektur pusat data bekerja pada tempat penyimpanan data pusat, baik aktif maupun pasif. Seperti kebanyakan skenario produsen-konsumen, produsen (bisnis) memproduksi barang ke penyimpanan data umum, dan konsumen (individu) dapat meminta data darinya. Terkadang, repositori pusat ini hanya berupa database sederhana.

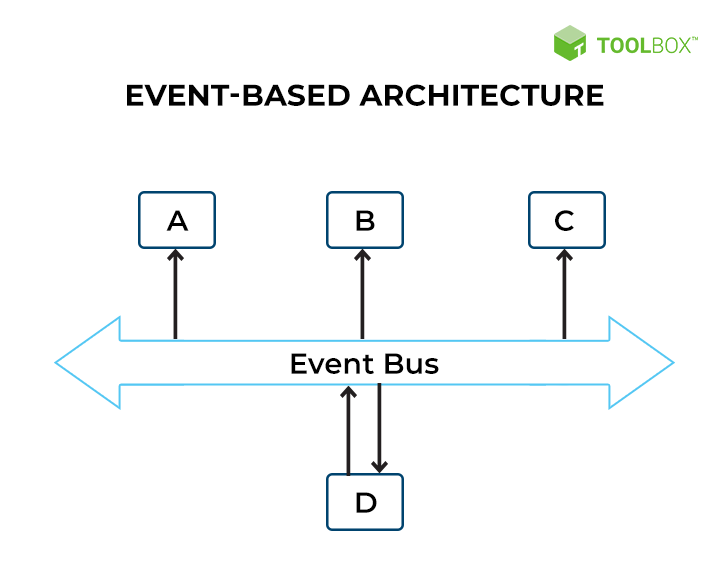
Semua komunikasi antar objek terjadi melalui sistem penyimpanan data dalam sistem yang berpusat pada data. Ini mendukung komponen penyimpanannya dengan ruang penyimpanan persisten seperti database SQL, dan sistem menyimpan semua node dalam penyimpanan data ini.



1. Arsitektur berbasis peristiwa (Event-based architecture)

Dalam arsitektur berbasis peristiwa, seluruh komunikasi dilakukan melalui peristiwa. Ketika suatu peristiwa terjadi, sistem mendapat notifikasi. Artinya siapapun yang menerima event ini juga akan diberitahu dan mempunyai akses informasi. Terkadang, peristiwa ini berupa data, dan terkadang berupa URL ke sumber daya. Dengan demikian, penerima dapat memproses informasi yang mereka terima dan bertindak sesuai dengan itu.

Salah satu keuntungan signifikan dari arsitektur berbasis peristiwa adalah komponen-komponennya digabungkan secara longgar. Pada akhirnya, ini berarti mudah untuk menambah, menghapus, dan memodifikasinya. Untuk lebih memahami hal ini, pikirkan sistem penerbit-pelanggan, bus layanan perusahaan, atau akka.io. Salah satu keuntungan dari arsitektur berbasis peristiwa adalah memungkinkan komponen heterogen untuk berkomunikasi dengan bus, apa pun protokol komunikasinya.



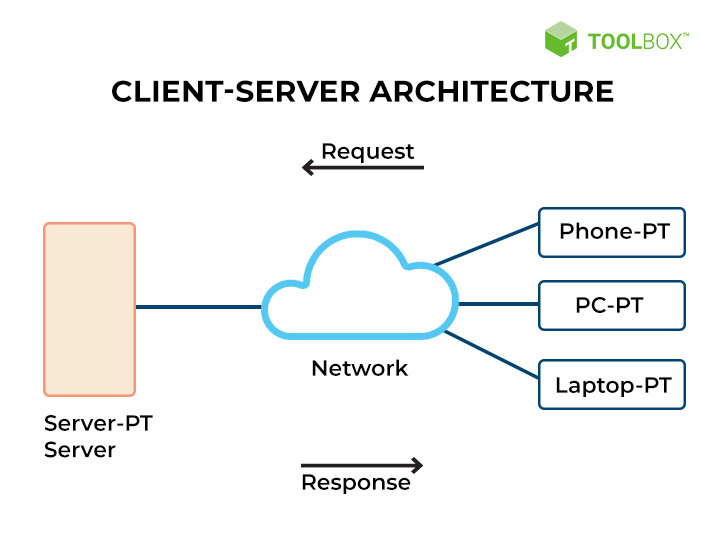
1. Arsitektur sistem

Arsitektur tingkat sistem berfokus pada keseluruhan sistem dan penempatan komponen sistem terdistribusi di beberapa mesin. Arsitektur client-server dan arsitektur peer-to-peer adalah dua arsitektur tingkat sistem utama yang memiliki arti penting saat ini. Contohnya adalah sistem ecommerce yang berisi lapisan layanan, database, dan bagian depan web.

1. Arsitektur klien-server

Seperti namanya, arsitektur klien-server terdiri dari klien dan server. Server adalah tempat semua proses kerja berada, sedangkan klien adalah tempat pengguna berinteraksi dengan layanan dan sumber daya lainnya (server jarak jauh). Klien kemudian dapat meminta dari server, dan server akan meresponsnya. Biasanya, hanya satu server yang menangani sisi jarak jauh; namun, penggunaan beberapa server menjamin keamanan total.

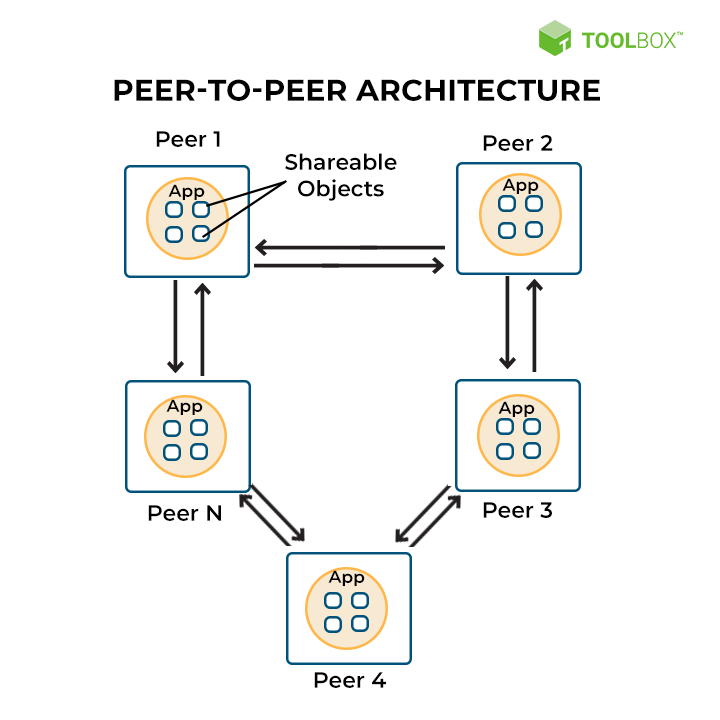
Arsitektur klien-server memiliki satu fitur desain standar: keamanan terpusat. Data seperti nama pengguna dan kata sandi disimpan dalam database yang aman agar pengguna server mana pun dapat mengakses informasi ini. Hal ini membuatnya lebih stabil dan aman dibandingkan peer-to-peer. Stabilitas ini berasal dari arsitektur client-server, di mana database keamanan dapat memungkinkan penggunaan sumber daya dengan cara yang lebih bermakna. Sistemnya jauh lebih stabil dan aman, meski tidak secepat server. Kerugian dari sistem terdistribusi adalah satu titik kegagalan dan tidak dapat diskalakan seperti server.



1. Arsitektur peer-to-peer (P2P).

Jaringan peer-to-peer, juga disebut jaringan (P2P), bekerja berdasarkan konsep tidak adanya kendali pusat dalam sistem terdistribusi. Sebuah node dapat bertindak sebagai klien atau server kapan saja setelah bergabung dengan jaringan. Node yang meminta sesuatu disebut klien, dan simpul yang menyediakan sesuatu disebut server. Secara umum, setiap node disebut peer.

Jika sebuah node baru ingin menyediakan layanan, ia dapat melakukannya dengan dua cara. Salah satu caranya adalah dengan mendaftar ke server pencarian terpusat, yang kemudian akan mengarahkan node ke penyedia layanan. Cara lainnya adalah node tersebut menyiarkan permintaan layanannya ke setiap node lain di jaringan, dan node mana pun yang merespons akan menyediakan layanan yang diminta.



Jaringan P2P saat ini memiliki tiga bagian terpisah:

* P2P Terstruktur: Node dalam P2P terstruktur mengikuti struktur data terdistribusi yang telah ditentukan sebelumnya.
* P2P Tidak Terstruktur: Node dalam P2P tidak terstruktur memilih tetangganya secara acak.
* P2P Hibrid: Dalam P2P hibrid, beberapa node memiliki fungsi unik yang ditetapkan secara berurutan.