**Nama : Alif Zidan Andriansyah**

**Kelas : TRPL 3A**

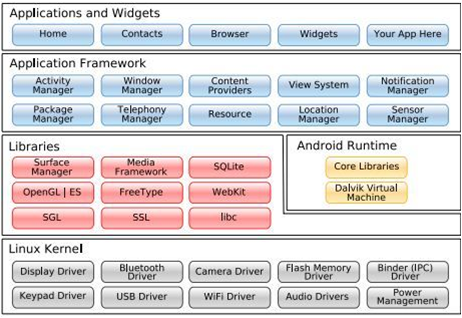
**NoBP : 2111083004**

**Arsitektur Sistem Terdistribusi**

**Arsitektur**

Definisi 🡪 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.

Arsitektur sistem terdistribusi 🡪 Arsitektur sistem terdistribusi merujuk pada desain dan organisasi suatu sistem komputer yang terdiri dari beberapa komponen atau node yang terdistribusi di lokasi yang berbeda, tetapi saling bekerja sama untuk mencapai tujuan tertentu.

[](https://1.bp.blogspot.com/-tdraNKbd1LE/V2o3sMysMcI/AAAAAAAAAeY/JTQ0wqyAy4MncEa91PYByIkB_NNoyHmkQCLcB/s1600/Arsitektur-1-ilmu-skripsi.png)

**Model Arsitektur Sistem Terdistribusi**

1. Arsitektur Logis(Software Architecture)

* Organisasi logika dari komponen-komponen perangkat lunak
* Komponen yang dimaksud berupa unit modular berupa interface yang dapat diproses di sistem yang berbeda
* RPC (remote prosedure call), message passing
* Jenis Model arsitektur logis (style)

🡪 Layered architectures

🡪 Object-base architectures

🡪 Data-Center architectures

🡪 Event-based architectures

1. Arsitektur Fisik(System Architecture)

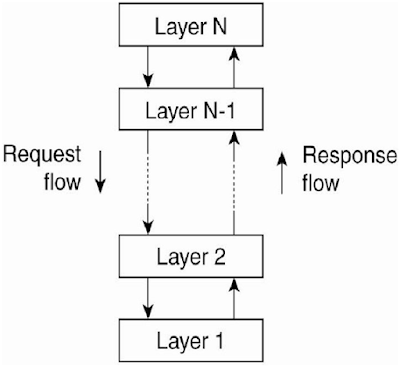
* Peletakan mesin
* Peletakan komponen perangkat lunak pada mesin sesungguhnya

**Layered Architectures**

Komponen-komponen pada Layered architectures diorganisasi dalam bentuk lapisan-lapisan (layer) fungsi dan service.

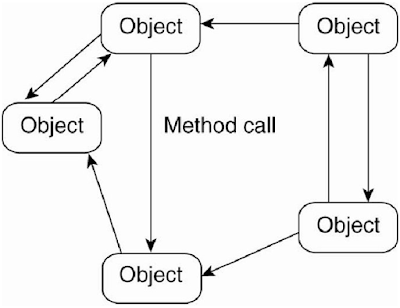
Contoh:

* Operating system (windows, linux)
* Network Protocol (OSI, TCP/IP)

[](https://4.bp.blogspot.com/-KDrSE3TTJ6s/V2o3uD-3WEI/AAAAAAAAAfA/GcdvZdWxGEEqfjZuULKYwn4bV0_sjz_YwCLcB/s1600/Arsitektur-2-ilmu-skripsi.png)

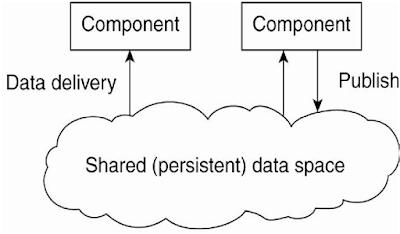
**Object-base Architectures**

Object-base architectures menggambarkan setiap objek melakukan koresponden dengan komponen, dan komponen ini terkoneksi melalui mekanisme procedure call. Bentuk sistem OA ini digunakan aplikasi perangkat lunak dalam skala besar.

[](https://3.bp.blogspot.com/-W6O1igfwLh4/V2o3uE01_LI/AAAAAAAAAfE/78mmS8xTri8XhgfOkTrdbKKJX52deJkoACLcB/s1600/Arsitektur-3-ilmu-skripsi.png)

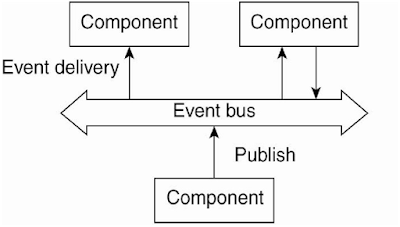
**Data-center Architectures**

Data center dapat dipandang sebagai gudang data (data warehouse) yang berfungsi sebagai sistem pengelolaan data mulai dari pengumpulan, pengolahan, penyimpanan hingga penemuan kembali data, serta mampu pula memberikan dukungan dalam pengambilan keputusan. Sebagai contoh adalah sistem tersebar

[](https://3.bp.blogspot.com/-pzcIvU7wT9g/V2o3uchQU0I/AAAAAAAAAfM/Yyrls3oyFaQbVnOtPqllvsAElypCug_4ACLcB/s1600/Arsitektur-4-ilmu-skripsi.png)

**Event-based Architectures**

Proses EBA pada dasarnya berdasarkan propagasi event. Proses mengeluarkan event setelah Middleware memberikan kepastian hanya proses itu saja yang bisa di subcribe untuk event yang diterima. Keuntungan EBA adalah proses bersifat loosely coupled.

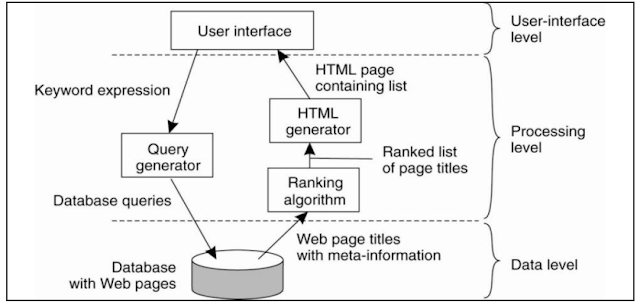
[](https://3.bp.blogspot.com/-0etr20RoMoE/V2o3uw1FaoI/AAAAAAAAAfU/rPeXMwPnzMQBfZKy8HywMbyrfhMKvOzNACLcB/s1600/Arsitektur-5-ilmu-skripsi.png)

**System Architecture**

1. Centralized Architectures (Client-Server)

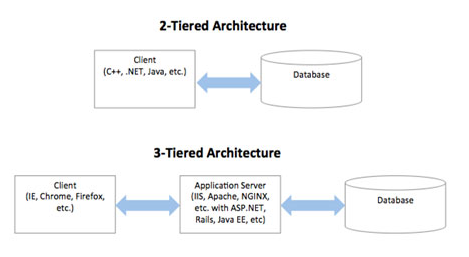
Client server unggul dalam kecepatan dan mendukung jaringan besar. Kekurangan terdapat pada sisi setup yang cukup komplek, biaya tinggi dan membutuhkan sumberdaya manusia yang handal untuk mengelola. Pada model client server, terdapat perilaku yang biasa disebut request-reply behaviour.

* Application Layering

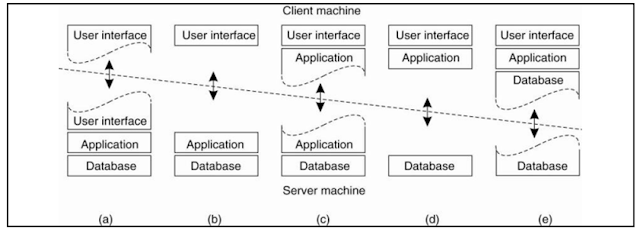
[](https://1.bp.blogspot.com/-2CvTu4Mu5kw/V2o3vilA1BI/AAAAAAAAAfg/PsNb91kwk7UOyvwaqL6w4a4oInw-eqCvgCLcB/s1600/Arsitektur-7-ilmu-skripsi.png)

Model client server seiring perkembangannya mengundang perdebatan mengenai perbedaan antara client dan server itu sendiri. Pada umumnya client server architecture ditujukan untuk keperluan user access ke database, maka dari itu layered architectural style dibagi menjadi :

* User-interface level (display management)
* Processing level (applications)
* Data level (actual data that is being acted on)
* Multi-tiered Architectures

[](https://4.bp.blogspot.com/-eEf9UqB1Ero/V2o3v7uikPI/AAAAAAAAAfk/lOEd3oI053Q1-vqhv0xjJYq56mDUnrnNgCLcB/s1600/Arsitektur-8-ilmu-skripsi.png)

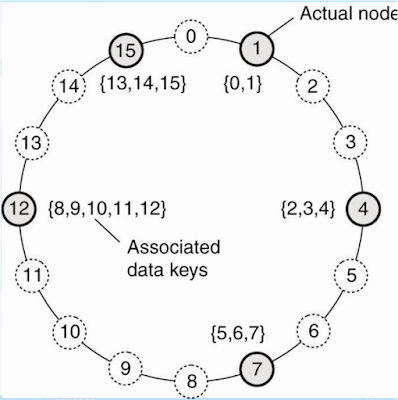
* 2 tier architecture
* Pengorganisasian paling simple dimana terdiri atas 2 type mesin
* Client yang berisi implementasi program pada user-interface level
* Server yang berisi implementasi program pada proses dan data level
* 3 tier architecture
* Pada keperluan khusus, kadang server juga perlu bertindak sebagai client
* Pada arsitektur ini, program pada processing level tidak hanya terdapat pada server yang terpisah, bahkan dapat terdistribusi pada client dan server mesin

[](https://4.bp.blogspot.com/-poOanwFOT1s/V2o3r00LK0I/AAAAAAAAAeQ/eAXAiFUNczkTcomYUvi_dvrk9nAB8qthQCLcB/s1600/Arsitektur-10-ilmu-skripsi.png)

* Trend perkembangan model arsitektur

1. Decentralized Architectures (Peer-to-Peer)

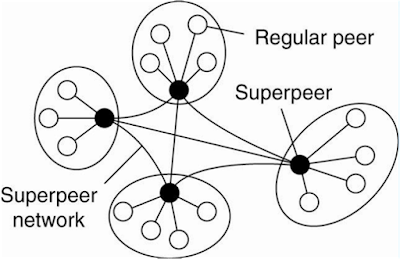
Disebut sebagai Horizontal distribution. Pada arsitektur ini, secara fisik terpisah namun secara logis memiliki fungsi level yang sama (equivalent), dimana setiap mesin memproses bagiannya sendiri kemudian melakukan balancing terhadap hasil proses.Nama lain Peer-to-peer architecture.

[](https://2.bp.blogspot.com/-ZlmsZ75lMk0/V2o3sE32b2I/AAAAAAAAAec/Qji9WFfR9nY9DrKJe9xujRbOm7Q6n6_pgCLcB/s1600/Arsitektur-11-ilmu-skripsi.png)

* Perkembangan P2P arsitektur tidak lepas dari pertanyaan

1. Bagaimana organisasi proses dalam jaringan
2. Sebuah proses tidak dapat berhubungan secara langsung dengan proses lain di jaringan.
3. Diperlukan sebuah pesan khusus untuk komunikasi proses

* Structured peer-to-peer architecture : Dalam struktur ini lapisan jaringan di bangun menggunakan deterministic procedure, seperti menggunakan distributes hash table (DHT).
* Unstructured peer-to-peer architecture : Dalam struktur ini menugaskan sebagian besar pada algoritma secara acak untuk membangun lapisan jaringan. Pada intinya setiap node mendata jaringan node neighboor, tetapi data node tersebut di tempuh dengan proses acak sederhana.
* Topology Management of Overlay Networks : Walaupun pada structured dan unstrustured peer-to-peer System cukup jelas, namun dalam beberapa kasus masih belum lengkap. Satu kunci dari observasi adalah kehati-hatian dari proses pertukaran dan pemilihan entries dari pandangan parsial dimana topologi tertentu dapat dibangun dan dijaga konektivitasnya.
* Superpeers

[](https://2.bp.blogspot.com/-vmxyDTrp_3E/V2o3swRxUzI/AAAAAAAAAek/3apxkoBsCqMbBo7ss08wnVNCqGO2lE4FgCLcB/s1600/Arsitektur-12-ilmu-skripsi.png)

1. Hybride Architectures

Edge-Server Systems. Sistem ini dibangun di jaringan internet dimana server di tempat kan pada edge (tepi) dari jaringan. Tujuan Edge server adalah melayani content (isi), pada saat proses filtering dan fungsi transcoding

**Contoh Arsitektur Sistem Terdistribusi**

1. Jaringan Sosial (Social Networking) :

* Facebook adalah contoh arsitektur sistem terdistribusi dalam konteks jaringan sosial.
* Dalam Facebook, pengguna berinteraksi dengan layanan melalui antarmuka web atau aplikasi seluler (klien).
* Server Facebook yang terdistribusi di seluruh dunia menyediakan data dan layanan yang diminta oleh pengguna.

1. Cloud Computing :

* Layanan komputasi awan seperti Amazon Web Services (AWS) dan Microsoft Azure adalah contoh arsitektur sistem terdistribusi.
* Penyedia awan menyediakan sumber daya komputasi (server, penyimpanan, jaringan) yang dapat diakses oleh pelanggan dari lokasi mana pun melalui internet.

1. Internet of Things (IoT) :

* Sistem IoT yang menghubungkan berbagai perangkat fisik seperti sensor, peralatan rumah tangga pintar, dan kendaraan otonom adalah contoh lain dari arsitektur sistem terdistribusi.
* Data dari perangkat IoT dikirimkan ke server pusat untuk diproses dan dianalisis.

1. Sistem Perbankan Terdistribusi:

* Sistem perbankan modern menggunakan arsitektur terdistribusi untuk mengelola transaksi dan menyediakan layanan perbankan online.
* ATM, situs web perbankan, dan aplikasi seluler adalah klien yang berkomunikasi dengan server bank yang terdistribusi.

**Kesimpulan**

Arsitektur sistem terdistribusi adalah suatu pendekatan desain dalam dunia komputasi yang memungkinkan komponen-komponen sistem tersebar di berbagai lokasi untuk bekerja bersama-sama dalam mencapai tujuan tertentu. Konsep utama melibatkan komunikasi efisien antar node, ketergantungan dan koordinasi tugas, serta fokus pada ketersediaan, toleransi kesalahan, dan skalabilitas. Dengan memanfaatkan prinsip-prinsip ini, arsitektur sistem terdistribusi memberikan keuntungan dalam meningkatkan kinerja, ketersediaan, dan fleksibilitas sistem, sehingga dapat mendukung aplikasi dan layanan yang memerlukan koordinasi dan pemrosesan terdistribusi.