

MODUL PEMBELAJARAN  
VEDULPRO

GAME DEVELOPER



## **Sisi klien**

Dalam pengembangan game, istilah "Sisi Klien" mengacu pada semua operasi dan aktivitas yang terjadi pada mesin pemain, yang dapat berupa konsol, komputer, atau bahkan telepon. Sisi klien bertanggung jawab untuk merender grafik, menangani masukan dari pengguna, dan terkadang memproses logika permainan. Hal ini berbeda dengan operasi sisi server, yang melibatkan penanganan koneksi multipemain dan sinkronisasi status permainan di antara banyak klien. Di sisi klien, pengembang perlu memastikan optimalisasi kinerja, UI/UX yang lancar, waktu muat yang cepat, dan keamanan untuk memberikan pengalaman bermain game yang menarik dan bebas lag. Keamanan juga penting untuk mencegah kecurangan dalam game multipemain, yang dapat diatasi melalui tindakan seperti kebingungan dan enkripsi data.

## **Sisi server**

Istilah "sisi server" mengacu pada operasi yang dilakukan oleh server dalam hubungan klien-server dalam jaringan komputer. Dalam permainan, sisi server bertanggung jawab atas logika dan aturan permainan, persistensi data, keamanan, dan sinkronisasi status permainan di antara semua klien yang terhubung. Ini pada dasarnya menangani semua pemrosesan yang tidak dilakukan di sisi klien. Kode yang dijalankan di sisi server dapat ditulis dalam berbagai bahasa pemrograman, namun sering kali dalam PHP, Ruby, Python, .NET, Java, atau JavaScript (Node.js). Mengetahui pemrograman sisi server sangat penting bagi pengembang game mana pun untuk membuat game online yang dapat dikelola, terukur, dan aman.

## **TCP**

TCP, singkatan dari Transmisi Control Protocol, adalah protokol dasar dalam rangkaian protokol internet. Ini adalah layanan aliran byte yang berorientasi koneksi dan andal yang berada di lapisan transport Model OSI. Dalam istilah yang lebih sederhana, TCP memungkinkan dua host untuk membuat koneksi dan bertukar data. Selama komunikasi ini, TCP memastikan bahwa data tidak hilang atau diterima secara tidak semestinya dengan menyediakan mekanisme pengecekan kesalahan dan sistem untuk transmisi ulang data yang hilang. Selain itu, untuk memastikan penggunaan sumber daya jaringan secara efisien, TCP menerapkan kontrol aliran, kontrol kemacetan, dan menyediakan sarana bagi host untuk menyesuaikan throughput koneksi secara dinamis.

# Jumlah pemeriksaan

Checksum di TCP (Transmission Control Protocol) adalah bidang 16 - bit di header TCP yang digunakan untuk melakukan pemeriksaan kesalahan pada segmen. Tumpukan TCP menghitung nilai checksum untuk data yang dikirimkan dan mengirimkannya bersama data ke sistem penerima. Sistem penerima menghitung ulang checksum dan membandingkannya dengan nilai yang dikirim bersama data. Jika kedua nilai yang dihitung cocok, data diasumsikan bebas dari kesalahan transmisi. Namun, jika nilai yang dihitung tidak cocok, TCP akan mendeteksi kemungkinan perubahan pada data yang diterima, dan penerima akan meminta transmisi ulang paket data yang hilang atau rusak. Harap dicatat bahwa meskipun operasi checksum membantu memastikan integritas data, operasi ini tidak sepenuhnya aman karena mungkin tidak mendeteksi semua kemungkinan kesalahan, terutama yang melibatkan perubahan beberapa bit.

## Struktur Segmen

TCP (Transmission Control Protocol) menggunakan metode yang disebut "segmentasi" untuk mengatur transmisi data. Dalam pendekatan ini, TCP membagi aliran data menjadi bagian-bagian yang lebih kecil, yang dikenal sebagai "segmen". Masing-masing segmen ini kemudian dikirim secara independen melalui internet. Segmen TCP dasar terdiri dari header dan bagian data. Header berisi berbagai bidang seperti port sumber, port tujuan, nomor urut, nomor pengakuan, offset data, bagian yang dipesan, bit kontrol, jendela, checksum, penunjuk mendesak, dan opsi. Segmen lainnya ditempati oleh data yang ditransfer melalui TCP. Memahami struktur segmen ini sangat penting untuk memahami bagaimana TCP, dan sebagian besar Internet, beroperasi.

## Bahasa pemrograman

Bahasa pemrograman adalah bagian penting dari pengembangan game. Mereka menciptakan logika dan aturan bagaimana game Anda beroperasi. Ada berbagai jenis bahasa pemrograman yang dapat digunakan oleh pengembang game sisi server, seperti C++, Python, Ruby, dll. Bahasa-bahasa ini memiliki sintaks dan struktur yang berbeda, serta tingkat kerumitan yang berbeda-beda. Kompleksitas dan kemampuan dapat memengaruhi performa, keamanan, dan bahkan potensi kompatibilitas game dengan platform lain. Pilihan bahasa Anda sering kali sangat bergantung pada kebutuhan spesifik proyek dan keahlian pribadi Anda. Mempelajari

bahasa baru bisa memakan waktu, jadi penting untuk memilih dengan bijak. Markdown adalah bahasa lain yang populer untuk dokumentasi karena kesederhanaan dan keterbacaannya, namun biasanya tidak digunakan untuk mengkodekan elemen game sisi server.

## Pemrograman Soket

Pemrograman soket adalah metode komunikasi antara dua komputer menggunakan protokol jaringan, biasanya TCP/IP. Dalam pengembangan game sisi server, pemrograman soket digunakan untuk memfasilitasi komunikasi dua arah secara real-time antara server dan klien. Server membuka soket yang menunggu permintaan koneksi masuk dari sisi klien. Setelah koneksi dibuat, server dan klien dapat bertukar data hingga koneksi ditutup. Hal ini memfasilitasi pengalaman bermain game multipemain secara langsung, memungkinkan pengguna di komputer berbeda untuk berinteraksi dalam lingkungan game yang sama. `socket programming` dalam Python, misalnya, menyertakan modul untuk membuat sisi server (`socketserver`) dan sisi klien (`socket`) percakapan.

## Serialisasi

Serialisasi dalam pengembangan game sisi server adalah proses mengubah objek atau struktur data ke dalam format yang nantinya dapat disimpan, dikirim, dan direkonstruksi. Ini penting karena memungkinkan Anda menyimpan status game, meneruskannya antar server, atau mengirimkannya ke klien. Dua jenis serialisasi utama adalah biner dan XML/JSON. Serialisasi biner mengubah objek Anda menjadi format biner yang sangat ringkas. Serialisasi XML/JSON mengubah objek Anda menjadi format tekstual, yang kurang efisien namun lebih mudah dibaca manusia dan lebih mudah untuk di-debug. Tergantung pada kebutuhan spesifik Anda, Anda dapat memilih salah satu dari yang lain. Perhatikan juga bahwa masing-masing metode ini memiliki kelebihan tersendiri dalam hal kompatibilitas, kinerja, dan implikasi keamanan.

## Multithread

Multithreading, suatu bentuk khusus dari multitasking, adalah kemampuan unit pemrosesan pusat (CPU) untuk mengelola beberapa eksekusi instruksi secara bersamaan. Ini pada dasarnya berarti bahwa beberapa thread atau proses mini dijalankan secara independen namun berbagi sumber daya dari satu CPU. Dalam

pemrograman, thread adalah cara untuk meningkatkan respons aplikasi dan melakukan banyak operasi secara bersamaan tanpa memerlukan banyak CPU atau komputer. Thread dalam suatu proses berbagi ruang data yang sama dengan thread utama dan oleh karena itu, dapat berkomunikasi lebih cepat satu sama lain dibandingkan jika mereka merupakan proses yang terpisah. Pengembang sering kali menggunakan multithreading dalam pengembangan game sisi server untuk mengelola operasi kompleks dengan efisiensi tinggi.

## **AI**

Kecerdasan Buatan (AI) dalam pengembangan game sisi server mengacu pada penggunaan algoritme dan prosedur komputasi untuk menciptakan sistem yang mampu melakukan tugas-tugas yang memerlukan kecerdasan manusia.

Tugas-tugas tersebut termasuk belajar dan beradaptasi terhadap perubahan, mengenali ucapan, atau bahkan membuat keputusan. Dalam pengembangan game, AI sering digunakan untuk memberikan 'kecerdasan' kepada karakter non-pemain (NPC), sehingga menjadikannya lebih hidup. Hal ini dapat berupa kemampuan sederhana untuk mengikuti pemain di suatu lingkungan, atau rumit seperti menyusun strategi untuk bertempur. AI juga dapat digunakan untuk menghasilkan konten yang dihasilkan secara prosedural atau dinamis, sehingga menciptakan pengalaman unik yang berpotensi tak terbatas bagi pemain. AI dalam game sering kali diprogram sedemikian rupa sehingga harus mencapai keseimbangan antara tampil cerdas dan memberikan pengalaman yang menyenangkan bagi pemain.

## **Permainan Matematika**

“Game Matematika” adalah aspek penting dari pengembangan game yang berhubungan dengan penggunaan konsep matematika untuk membuat dan mengontrol mekanisme game. Ini melibatkan bidang-bidang seperti geometri untuk pemodelan 3D, logika untuk aturan permainan, aljabar untuk sistem penilaian, dan trigonometri untuk pergerakan atau lintasan. Memahami matematika game memungkinkan pengembang untuk mengimplementasikan fitur seperti simulasi fisika, perilaku AI, dan pembuatan prosedural. Topik tingkat lanjut mencakup perhitungan kompleks untuk grafik (misalnya shader, pencahayaan) dan kalkulus untuk animasi berkelanjutan atau fisika tingkat lanjut. Kompleksitas matematis bergantung pada permintaan game, namun dasar yang kuat sangat penting bagi pengembang game mana pun.

## Fisika Permainan

*Fisika permainan* merupakan bagian integral dari pengembangan game yang mensimulasikan hukum fisika dalam lingkungan virtual. Simulasi ini menghadirkan realisme ke dalam permainan dengan mendefinisikan bagaimana objek bergerak, berinteraksi, dan bereaksi terhadap tumbukan dan gaya. Fisika permainan berkisar dari cara karakter melompat atau bergerak dalam ruang 2D atau 3D, hingga mekanika yang lebih kompleks seperti dinamika fluida atau fisika ragdoll. Dua jenis utama fisika permainan adalah 'fisika arcade', yang lebih sederhana dan abstrak; dan 'fisika realistis', yang berupaya menciptakan kembali interaksi fisika kehidupan nyata. Menerapkan fisika permainan memerlukan kombinasi pengetahuan matematika dan keterampilan pemrograman untuk mengintegrasikan mesin fisika seperti PhysX dari Unity dan alat fisika bawaan Unreal Engine.

## Mesin Permainan

*Game Engine* adalah kerangka perangkat lunak yang dirancang untuk memfasilitasi pembuatan dan pengembangan video game. Pengembang menggunakannya untuk membuat game untuk konsol, perangkat seluler, dan komputer pribadi. Fungsi inti yang biasanya disediakan oleh mesin permainan mencakup mesin rendering ("renderer") untuk grafik 2D atau 3D, mesin fisika atau deteksi tabrakan (dan respons tabrakan), suara, skrip, animasi, kecerdasan buatan, jaringan, streaming, memori manajemen, dan grafik adegan. Game Engine dapat menghemat banyak waktu pengembangan dengan menyediakan komponen yang dapat digunakan kembali ini. Namun, solusi tersebut bukanlah solusi yang bisa diterapkan untuk semua orang, karena pengembang masih harus menyesuaikan sebagian besar kode agar sesuai dengan kebutuhan unik game mereka. Beberapa mesin permainan yang populer adalah Unity, Unreal Engine, dan Godot.

## API Grafik

Graphics API (Application Programming Interface) adalah kumpulan perintah, fungsi, protokol, dan alat yang digunakan pengembang game untuk membuat game. Ini membentuk antarmuka antara game dan perangkat keras perangkat, biasanya komputer atau konsol, dan membantu dalam merender kinerja grafis 2D dan 3D. Tugas kompleks seperti menggambar poligon, memberi tekstur, atau pencahayaan dikemas dalam proses tingkat tinggi yang lebih mudah dikelola oleh API. Contoh umum adalah Vulkan, DirectX, OpenGL, dan Metal. Masing-masing memiliki ketersediaan dan kinerja yang bervariasi di berbagai platform dan perangkat serta memiliki fitur unik yang dapat dimanfaatkan untuk pengembangan game.

## Permainan AI

Istilah "API" adalah singkatan dari Antarmuka Pemrograman Aplikasi. Secara sederhana, *Game API* adalah seperangkat protokol dan alat yang digunakan untuk membangun perangkat lunak dan aplikasi. Pada dasarnya, ini adalah metode bagi aplikasi perangkat lunak yang berbeda untuk berkomunikasi satu sama lain. Dalam hal pengembangan game, Game API menyediakan jembatan untuk berinteraksi dengan mesin game, memungkinkan pengembang untuk membuat skrip perilaku atau fungsi dalam game, seperti rendering grafis, penghitungan fisika, suara, dan AI (Kecerdasan Buatan). API game populer termasuk DirectX dan OpenGL. DirectX digunakan secara luas dalam pengembangan video game untuk Windows dan Xbox. OpenGL, di sisi lain, adalah API lintas platform untuk merender grafik 2D dan 3D dalam video game.

## Rendering Tingkat Lanjut

Rendering tingkat lanjut adalah teknik canggih yang digunakan dalam pengembangan game yang melibatkan penerjemahan model atau adegan 3D menjadi gambar atau animasi 2D. Teknik rendering tingkat lanjut dapat melibatkan berbagai metode kompleks seperti rendering berbasis fisik, penelusuran sinar, iluminasi global, hamburan bawah permukaan, kaustik, dan rendering volumetrik. Penggunaan rendering tingkat lanjut dapat menghasilkan grafik yang sangat realistis, karena menggunakan perhitungan rumit untuk menggambarkan perilaku cahaya di dunia nyata. Render tingkat lanjut sering kali memerlukan sumber daya perangkat keras yang kuat dan perangkat lunak khusus untuk mendapatkan gambar dan animasi yang diinginkan.