Set instruksi

**Set Instruksi** (bahasa [Inggris](https://id.wikipedia.org/wiki/Inggris): **Instruction Set**, atau **Instruction Set Architecture** (ISA)) didefinisikan sebagai suatu aspek dalam [arsitektur komputer](https://id.wikipedia.org/wiki/Arsitektur_komputer) yang dapat dilihat oleh para pemrogram. Secara umum, ISA ini mencakup jenis [data](https://id.wikipedia.org/wiki/Data) yang didukung, jenis [instruksi](https://id.wikipedia.org/w/index.php?title=Instruksi&action=edit&redlink=1) yang dipakai, jenis [register](https://id.wikipedia.org/wiki/Register), [mode pengalamatan](https://id.wikipedia.org/w/index.php?title=Mode_pengalamatan&action=edit&redlink=1), [arsitektur memori](https://id.wikipedia.org/w/index.php?title=Arsitektur_memori&action=edit&redlink=1), penanganan [interupsi](https://id.wikipedia.org/wiki/Interupsi), [eksepsi](https://id.wikipedia.org/w/index.php?title=Exception&action=edit&redlink=1), dan operasi I/O eksternalnya (jika ada).

* ISA merupakan sebuah spesifikasi dari Pullman semua kode-kode [biner](https://id.wikipedia.org/wiki/Biner) (*opcode*) yang diimplementasikan dalam bentuk aslinya (*native form*) dalam sebuah desain [prosesor](https://id.wikipedia.org/wiki/Mikroprosesor)tertentu. Kumpulan *opcode* tersebut, umumnya disebut sebagai **bahasa mesin** (*machine language*) untuk ISA yang bersangkutan. ISA yang populer digunakan adalah set instruksi untuk chip [Intel x86](https://id.wikipedia.org/wiki/X86), [IA-64](https://id.wikipedia.org/w/index.php?title=IA-64&action=edit&redlink=1), [IBM PowerPC](https://id.wikipedia.org/w/index.php?title=IBM_PowerPC&action=edit&redlink=1), [Motorola 68000](https://id.wikipedia.org/w/index.php?title=Motorola_68000&action=edit&redlink=1), [Sun SPARC](https://id.wikipedia.org/w/index.php?title=Sun_SPARC&action=edit&redlink=1), [DEC Alpha](https://id.wikipedia.org/wiki/Alpha_AXP), dan lain-lain.
* ISA kadang-kadang digunakan untuk membedakan kumpulan karakteristik yang disebut di atas dengan mikroarsitektur prosesor, yang merupakan kumpulan teknik desain prosesor untuk mengimplementasikan set instruksi (mencakup [*microcode*](https://id.wikipedia.org/w/index.php?title=Microcode&action=edit&redlink=1), [*pipeline*](https://id.wikipedia.org/wiki/Pipeline), [*sistem cache*](https://id.wikipedia.org/w/index.php?title=Sistem_cache&action=edit&redlink=1), [manajemen daya](https://id.wikipedia.org/w/index.php?title=Manajemen_daya&action=edit&redlink=1), dan lainnya). Komputer-komputer dengan[mikroarsitektur](https://id.wikipedia.org/w/index.php?title=Mikroarsitektur&action=edit&redlink=1) berbeda dapat saling berbagi set instruksi yang sama. Sebagai contoh, prosesor [Intel Pentium](https://id.wikipedia.org/wiki/Intel_Pentium) dan prosesor [AMD Athlon](https://id.wikipedia.org/wiki/AMD_Athlon) mengimplementasikan versi yang hampir identik dari set instruksi Intel x86, tetapi jika ditinjau dari desain internalnya, perbedaannya sangat radikal. Konsep ini dapat diperluas untuk ISA-ISA yang unik seperti TIMI yang terdapat dalam [IBM System/38](https://id.wikipedia.org/w/index.php?title=IBM_System/38&action=edit&redlink=1) dan [IBM IAS/400](https://id.wikipedia.org/w/index.php?title=IBM_IAS/400&action=edit&redlink=1). TIMI merupakan sebuah ISA yang diimplementasikan sebagai perangkat lunak level rendah yang berfungsi sebagai mesin virtual. TIMI didesain untuk meningkatkan masa hidup sebuah platform dan aplikasi yang ditulis untuknya, sehingga mengizinkan platform tersebut agar dapat dipindahkan ke perangkat keras yang sama sekali berbeda tanpa harus memodifikasi perangkat lunak (kecuali yang berkaitan dengan TIMI). Hal ini membuat IBM dapat memindahkan platform [AS/400](https://id.wikipedia.org/w/index.php?title=AS/400&action=edit&redlink=1) dari arsitektur mikroprosesor [CISC](https://id.wikipedia.org/wiki/CISC) ke arsitektur mikroprosesor [POWER](https://id.wikipedia.org/w/index.php?title=POWER&action=edit&redlink=1) tanpa harus menulis ulang bagian-bagian dari dalam sistem operasi atau perangkat lunak yang diasosiasikan dengannya.
* Ketika mendesain mikroarsitektur, para desainer menggunakan [Register Transfer Language](https://id.wikipedia.org/w/index.php?title=Register_Transfer_Language&action=edit&redlink=1) (RTL) untuk mendefinisikan operasi dari setiap instruksi yang terdapat dalam ISA.
* Sebuah ISA juga dapat diemulasikan dalam bentuk [perangkat lunak](https://id.wikipedia.org/wiki/Perangkat_lunak) oleh sebuah [interpreter](https://id.wikipedia.org/wiki/Interpreter). Karena terjadi translasi tambahan yang dibutuhkan untuk melakukan [emulasi](https://id.wikipedia.org/w/index.php?title=Emulasi&action=edit&redlink=1), hal ini memang menjadikannya lebih lambat jika dibandingkan dengan menjalankan program secara langsung di atas perangkat keras yang mengimplementasikan ISA tersebut. Akhir-akhir ini, banyak vendor ISA atau mikroarsitektur yang baru membuat perangkat lunak emulator yang dapat digunakan oleh para pengembang perangkat lunak sebelum implementasi dalam bentuk [perangkat keras](https://id.wikipedia.org/wiki/Perangkat_keras) dirilis oleh vendor.

**Bagian dari instruksi**

[](https://id.wikipedia.org/wiki/Berkas:Mips32_addi.svg)

Satu instruksi mungkin memiliki beberapa bidang, yang mengidentifikasi operasi logis untuk dilakukan, dan bisa juga menyertakan alamat sumber dan tujuan dan nilai-nilai konstan. Ini adalah MIPS "Tambahkan Segera" instruksi yang memungkinkan pemilihan register sumber dan tujuan dan inklusi dari sebuah konstanta kecil. pada arsitektur tradisional, instruksi mencakup [opcode](https://id.wikipedia.org/w/index.php?title=Opcode&action=edit&redlink=1)menentukan operasi yang akan dilakukan, seperti "isi menambah memori untuk mendaftar", dan nol atau lebih [operan](https://id.wikipedia.org/w/index.php?title=Operan&action=edit&redlink=1) penspesifikasi, yang dapat menentukan [register](https://id.wikipedia.org/w/index.php?title=Prosesor_register&action=edit&redlink=1), memori lokasi, atau data harfiah. Para penspesifikasi operan mungkin memiliki [mode pengamatan](https://id.wikipedia.org/w/index.php?title=Mode_pengamatan&action=edit&redlink=1)menentukan makna mereka atau mungkin dalam bidang tetap. Dalam [kata sangat panjang instruksi [

(VLIW) arsitektur, yang mencakup banyak [microcode](https://id.wikipedia.org/w/index.php?title=Microcode&action=edit&redlink=1) arsitektur, opkode simultan dan operan yang ditentukan dalam sebuah instruksi.

Beberapa set instruksi eksotis tidak memiliki bidang opcode (seperti [Transportasi Dipicu Arsitektur](https://id.wikipedia.org/w/index.php?title=Transportasi_Dipicu_Arsitektur&action=edit&redlink=1) (TTA) atau [mesin Forth maya](https://id.wikipedia.org/w/index.php?title=Mesin_Forth_maya&action=edit&redlink=1)), hanya operan (s). Lainnya tidak biasa " [0-operan](https://id.wikipedia.org/w/index.php?title=Instruksi_set&action=edit&redlink=1)" set instruksi kekurangan dalam suatu operan bidang specifier, seperti beberapa [mesin [tumpukan]] termasuk [NOSC](https://id.wikipedia.org/w/index.php?title=NOSC&action=edit&redlink=1) [/ balik / NOSC /](http://strangegizmo.com/).

ISA yang diimplementasikan dalam bentuk perangkat keras

* [Alpha AXP](https://id.wikipedia.org/wiki/Alpha_AXP) (DEC Alpha)
* [ARM](https://id.wikipedia.org/w/index.php?title=ARM&action=edit&redlink=1) (Acorn RISC Machine) (Advanced RISC Machine now ARM Ltd)
* [IA-64](https://id.wikipedia.org/w/index.php?title=IA-64&action=edit&redlink=1) ([Itanium](https://id.wikipedia.org/wiki/Itanium)/[Itanium 2](https://id.wikipedia.org/w/index.php?title=Itanium_2&action=edit&redlink=1))
* [MIPS](https://id.wikipedia.org/wiki/MIPS)
* [Motorola 68k](https://id.wikipedia.org/w/index.php?title=Motorola_68000&action=edit&redlink=1)
* [PA-RISC](https://id.wikipedia.org/w/index.php?title=PA-RISC&action=edit&redlink=1) (HP Precision Architecture)
* [IBM POWER](https://id.wikipedia.org/w/index.php?title=POWER&action=edit&redlink=1)
* [IBM PowerPC](https://id.wikipedia.org/w/index.php?title=IBM_PowerPC&action=edit&redlink=1)
* [SPARC](https://id.wikipedia.org/wiki/SPARC)
* [SuperH](https://id.wikipedia.org/w/index.php?title=SuperH&action=edit&redlink=1) (Hitachi)
* [System/360](https://id.wikipedia.org/w/index.php?title=System/360&action=edit&redlink=1)
* Tricore (Infineon)
* Transputer (STMicroelectronics)
* [VAX](https://id.wikipedia.org/w/index.php?title=VAX&action=edit&redlink=1) (Digital Equipment Corporation)
* [x86](https://id.wikipedia.org/wiki/X86) (IA-32, [Pentium](https://id.wikipedia.org/wiki/Pentium), [Athlon](https://id.wikipedia.org/wiki/Athlon)) ([AMD64](https://id.wikipedia.org/wiki/AMD64), [EM64T](https://id.wikipedia.org/w/index.php?title=EM64T&action=edit&redlink=1))

ISA yang diimplementasikan dalam bentuk perangkat lunak lalu dibuat perangkat kerasnya

* p-Code (UCSD p-System Version III on Western Digital Pascal Micro-Engine)
* [Java virtual machine](https://id.wikipedia.org/wiki/Java) (ARM Jazelle, PicoJava)
* FORTH

ISA yang tidak pernah diimplementasikan dalam bentuk perangkat keras

* SECD machine
* ALGOL Object Code