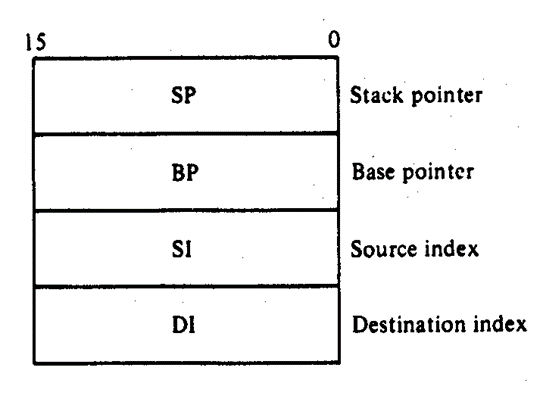


Sumber:  https://imgur.com/RFGA4nz

Microprocessor 8088 mempunyai 4 bagian register utama, yaitu

1. register alamat (Pointer and Index Register)
2. register fungsi data (general purpose register)
3. register segment (segment register),
4. register tanda (flag register).

**A. Register alamat**



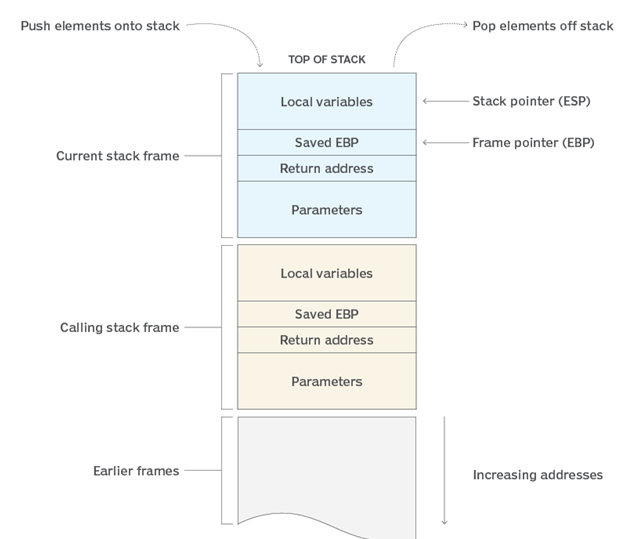
**Sumber:** https://slideplayer.com/slide/3944489/

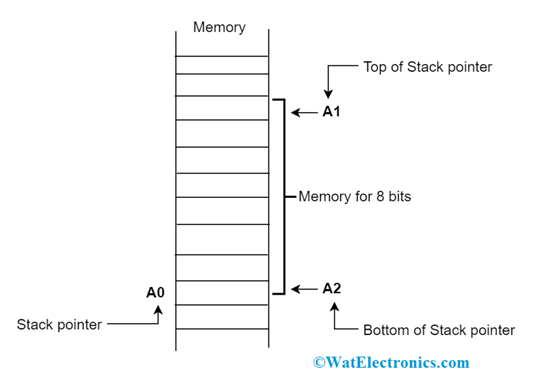
Register penunjuk intruksi (Bahasa Inggris: Pointer and Index Register) merupakan register 16 bit berisikan offset data dan intruksi yang digunakan untuk menunjukkan lokasi instruksi berikutnya yang akan dijalankan. Nilai yang disimpan dalam register ini berada di alamat Offset yang mengacu pada jarak variable, label, atau intruksi dari base segment. Register ini digunakan dengan data tersusun (seperti string, array dan struktur data lain yang berisi banyak elemen). Register ini membantu meningkatkan kecepatan pemrosesan struktur data tersebut. Register ini digunakan untuk aritmatika dan lainnya.

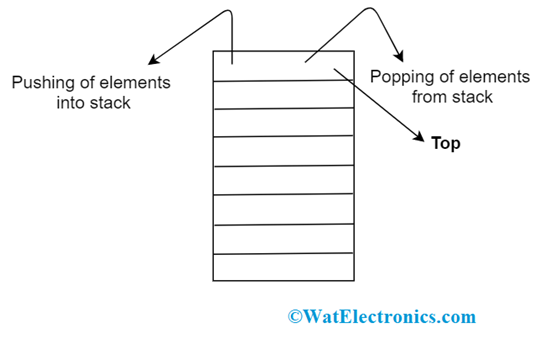
Register ini berpasangan dengan Code Segment dengan penulisannya [CS:IP] di mana lokasi alamat yang ditunjuk bergantung pada code segment yang terdapat pada segment register. Kita bisa juga menggabungkan SP dengan nilai di dalam SS (SS:SP) menghasilkan  alamat 20-bit yang mengarah ke stack segment/top of the stack (TOS) yang berada di RAM.

Dari register penunjuk intruksi, terbagi menjadi dua bagian register yaitu register pointer yang mempunyai dua buah register penunjuk (register SP dan BP) serta register dua buah register indeks (register SI dan DI).

1. **Stack Pointer (SP)**







Sumber:

**https://www.watelectronics.com/why-stack-and-stack-pointer-matters-the-most/**

Stack Pointer (SP) adalah register 16-bit menunjuk ke stack segment yang menyimpan alamat memori dari elemen data terakhir yang ditambahkan untuk mendukung subroutine calls dan interrupt/exception. Stack adalah buffer khusus yang digunakan oleh fungsi program untuk menyimpan data seperti parameter, variable lokal, dan informasi terkait lainnya. Data yang terdapat pada SP memungkinkan pemogram untuk menjangkau lokasi alamat offset dari stack segment.

Tata letak SP bergantung pada bahasa pemrograman dan arsitektur processor 8088. Secara umum, prosesor 8088 menyimpan SPfsac di register perangkat keras dan ALU (Arithmatic and Logic Unit) berfungsi untuk menghitung nilai. Biasanya di assembly ada perintah mnemonic push dan pop yang dapat mengubah sebagai operasi kecil untuk menambahkan atau menghapus SP secara individual dan melakukan pemuatan dan penyimpanan dalam memori.

Cara kerja SP yaitu Ketika permintaan baru tiba kemudian akan menekan permintaan sebelumnya. Jadi permintaan terbaru selalu ditempatkan di posisi teratas stack dan program akan mendapatkan permintaan hanya dari posisi teratas. Dengan masuknya permintaan yang baru, maka stack pointer bergerak maju ke alamat memori fisik berikutnya dan elemen terbaru direplikasi di lokasi alamat baru. Dengan cara yang sama, Ketika sebuah elemen dihapus dari stack, SP Kembali ke lokasi berikutnya yang ada di posisi teratas dalam stack.

**2. Base Pointer (BP)**

Base Pointer (BP) adalah register 16-bit yang menunjuk ke data dalam stack segment. Register BP biasanya digunakan sebagai based, indexed atau register indirect yang hal ini digunakan sebagai referensi parameter.

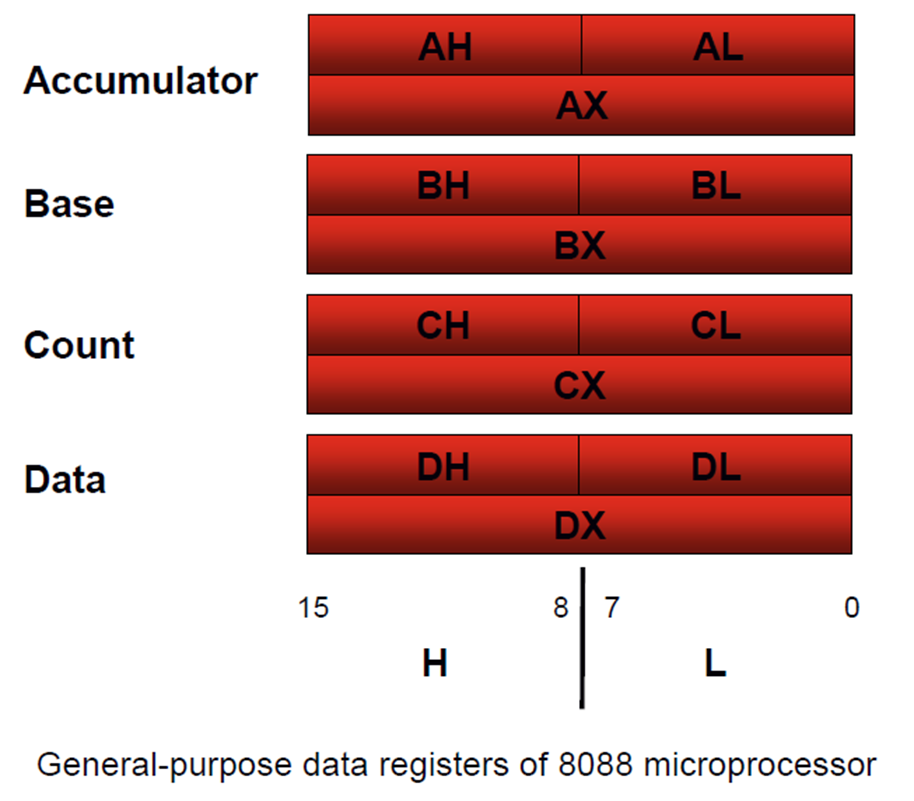
**3. Source Index (SI)**

Source Index (SI) adalah 16-bit register digunakan untuk indexed, based indexed, dan pengalamatan register tidak langsung, dimana sebagai alamat data sumber dalam intruksi memanipulasi string.

**4. Destination Index (DI)**

Destination Index (DI) adalah 16-bit register digunakan pada operasi string dengan mengakses secara langsung pada alamat tujuan memori yang ditunjukkan.

**B. Register Fungsi Data**



**Sumber:** https://slideplayer.com/slide/3944489/

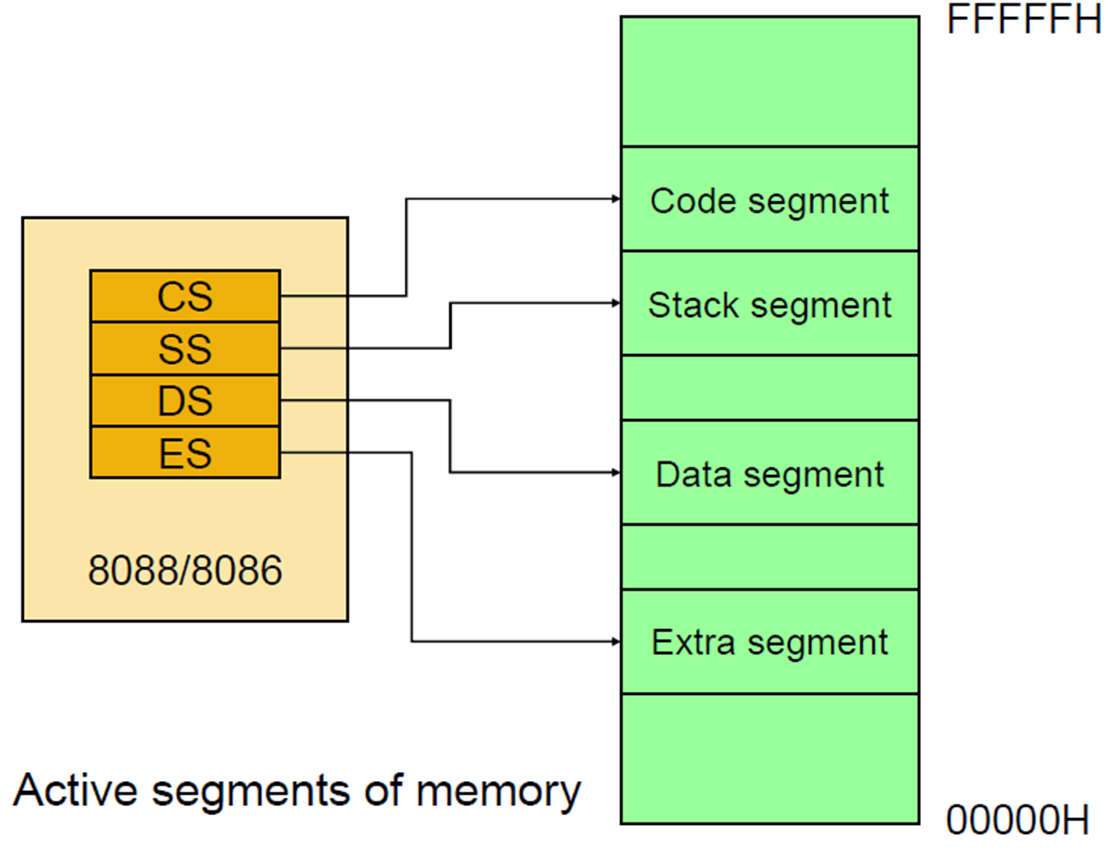
Register fungsi Data (Bahasa inggris: general purpose register) adalah register yang berfungsi untuk tempat penyimpanan data secara sementara. Register ini memiliki lebar 16 bit tetapi dapat mempunyai 2 bagian dimana masing-masing terdiri dari 8 bit.

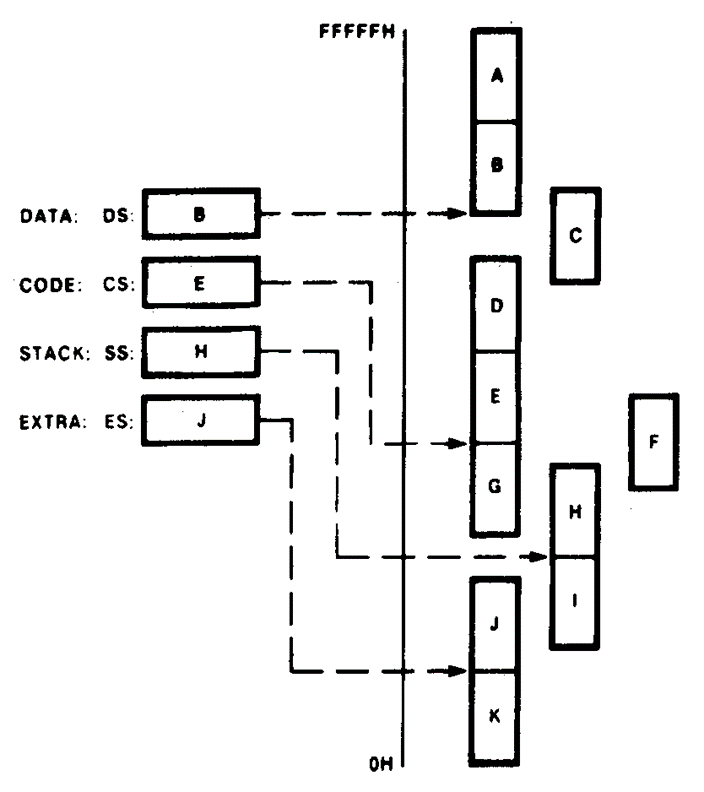
|  |  |  |
| --- | --- | --- |
|  | **Lebar data** | |
| 8 Bit | 16 Bit |
| **Accumulator** | AH,AL | AX |
| **Base Register** | BH,BL | BX |
| **Counter Register** | CH,CL | CX |
| **Data Segment** | DH,DL | DX |

Berikut kegunaan register fungsi data pada masing-masing register

|  |  |
| --- | --- |
| **Register** | **Operas**i |
| AX | Perkalian word, Pembagian word, I/O word |
| AL | Perkalian byte, pembagian byte, I/O byte, keterangan, aritmatika decimal |
| AH | Perkalian byte, pembagian btyte |
| BX | Keterangan |
| CX | Operasi string, perulangan (loop) |
| CL | Variable shift dan rotate |
| DX | Perkalian word, pembagian word, I/O secara tidak langsung |

**C. Register Segment**





**Sumber**: https://slideplayer.com/slide/3944489/

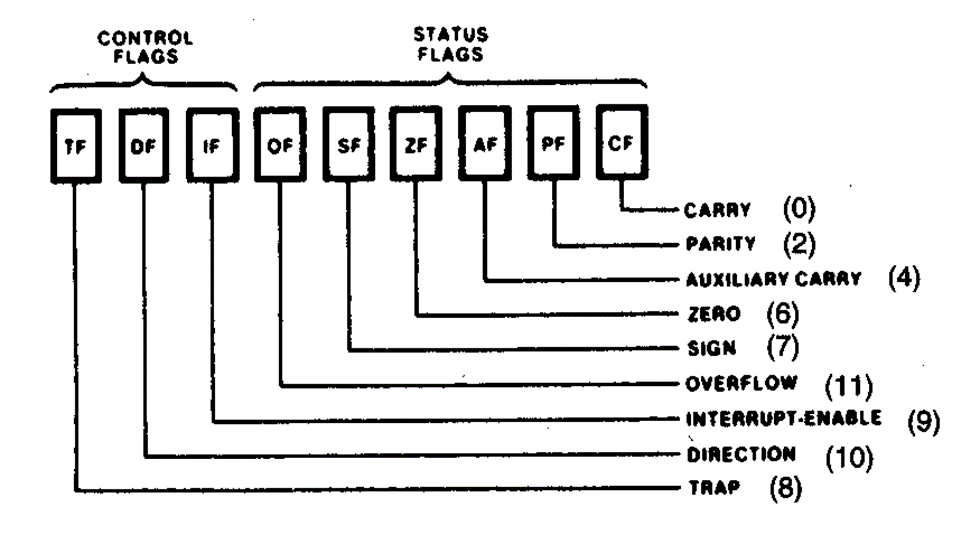
Register segmen (bahasa inggris: segment register) adalah register 16-bit dimana mewakili unit memori yang dapat dialamatkan secara independen yang terdiri dari 64KB ( 216 = 64 KB), lebar penyimpanan byte berturut-turut. Pada Mikroprosesor 8088 menghasilkan 20 bit alamat sehingga dapat menjangkau 1 MB lokasi memori ( 220 = 1048576 Byte atau disingkat 1 MB).  sehingga untuk menjangkau alamat 1 MB, memori pada µP 8088 dibagi menjadi ruas-ruas 64 KB, sehingga setiap saat hanya ada 4 segment (ruas) yang dapat aktif. Segment-segment register ini adalah:

1. Code Segment (CS), digunakan untuk menyimpan program.
2. Data Segment (DS), digunakan untuk menyimpan data program.
3. Stack Segment (SS), untuk menyimpan segment data tambahan.
4. Extra Segment (ES), Stack segment digunakan nuntuk menyimpan alamat kembalinya interupsi dan subrutin.

Keempat register diatas menunjuk ke lokasi masing-masing segment. Karena register segment hanya terdiri dari 16 bit, sedangkan memori yang dapat dijangkau sebanyak 20 bit, maka unit BIU ( Bus Interface Unit) akan menambah 4 bit lagi pada LSB.

Misalnya jika register ES=2721H, maka register akan menunjuk kelokasi 27210H. Setiap segment hanya dapat menampung 64 KB lokasi memori, sehingga jika CS=A000H, maka lokasi memori untuk Code Segment dimulai dari A0000H sampai AFFFFH (64 KB).

**D. Register Tanda**



**Sumber:**  https://slideplayer.com/slide/3944489/

Register tanda (bahasa inggris: Flag Register)  merupakan register 16 bit, namun pada mikroprocessoor 8088 yang digunakan hanya 9 bit, yaitu

* 6 dari bit digunakan status.
* 3 dari bit digunakan sebagai kontrol..

Microprocessor 8088 menyediakan instruksi di dalam set instruksinya yang dapat menggunakan flag ini untuk mengubah urutan di mana program dijalankan.

Bentuk diagram blok pada flag register adalah sebagai berikut :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | X | X | X | OF | DF | IF | TF | SF | ZF | X | AF | X | PF | X | CF |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

 Berikut fungsi masing-masing register tersebut.

|  |  |
| --- | --- |
| CF (Carry Flag) | jika berlogic 1 berarti terdapat carry atau borrow pada MSB (Most Significant Bit) yang terjadi selama operasi aritmatika. Jika berlogic 0 berarti tidak terdapat carry atau borrow. |
| PF (Parity Flag) | jika berlogic 1 berarti 8 bit terendah menghasilkan paritas genap dan jika manghasilkan paritas ganjil PF akan berlogic 0. |
| AF (Auxillary Carry Flag) | jika berlogic 1 berarti terdapat carry pada bit ke 4 pada register AL dan bila tidak akan berlogic 0. Register ini biasanya digunakan pada operasi BCD, seperti perintah AAA. ZF (Zero Flag), jika berlogic 1 maka operasi aritmatika menghasikan sisa 0, jika berlogic 1 maka tidak menghasilkan 0 |
| SF (Sign Flag) | jika digunakan bilangan bertanda bit ini akan bernilai 1. Sedangkan bila SF berlogic 0 berarti bilangan diperlakukan sebagai bilangan tidak bertanda. Bilangan bertanda dibagi menjadi bilangan positif (+) dan bilangan negatif (-). Pada bilangan bertanda, bit terakhir (bit ke-16) diperlakukan sebagai tanda (+) atau tanda(-). Jika bit terakhir tersebut bernilai 1 berarti bilangan tersebut negatif dan jika bit terakhir bernilai 0 berarti bilangan tersebut positif |
| TF (Trace Flag) | jika berlogic 1 berarti berada pada keadaan single step. Keadaan ini digunakan pada program Debug |
| IF (Interrupt Flag) | jika berlogic 1 berarti Maskable Interrupt Request dapat dilakukan. Jika berlogic 0 maka permintaan interupsi tidak dapat dipenuhi oleh CPU |
| OF (Over Flow Flag) | jika terjadi Over Flow pada operasi aritmatika, bit ini akan bernilai 1. Dan jika tidak terjadi Over Flow pada operasi aritmatika, bit ini akan bernilai 0 |
| DF (Direction Flag) | jika berlogic 1 berarti pada instruksi string nilai register akan diturunkan secara otomatis dan jika berlogic 0 maka akan dinaikkan secara otomatis |
| X | Tidak digunakan |