MENCARI DAFTAR INTRUKSI SET INTEL.

Petunjuk baru di Processor Intel486 ™  
  
Petunjuk berikut adalah baru dalam prosesor Intel486:  
• BSWAP (byte swap) instruksi.  
• XADD (pertukaran dan tambahkan) instruksi.  
• CMPXCHG (bandingkan dan pertukaran) instruksi.  
• NVD (membatalkan cache) instruksi.  
• WBINVD (write-back dan membatalkan cache) instruksi.  
• INVLPG (membatalkan TLB entri) instruksi.

**Intruksi set 8085**

Untuk dapat menggunakan suatu mikroprosesor, maka harus mengetahui set instruksi dari mikroprosesor yang bersangkutan. Set instruksi dari suatu mikroprosesor berisikan daftar seluruh instruksi-instruksi yang dapat dilakukan oleh mikroprosesor tersebut. Pada mikroprosesor 8085 terdapat lebih dari 70 instruksi yang terbagi menjadi beberapa kelompok instruksi, yaitu :

1.      Kelompok instruksi transfer data

2.      Kelompok instruksi aritmatika

3.      Kelompok instruksi geser (“shift” dan “rotate”)

4.      Kelompok instruksi logika

5.      Kelompok instruksi pencabangan (“branch)

6.      Kelompok instruksi input/output dan control

Kelompok Instruksi :

**1. Transfer Data**

Di antara semua kelompok instruksi pada mikroprosesor 8085, kelompok instruksi transfer data menunjukkan bagaimana cara memindahkan data 8 bit dari suatu register ke register lain atau antara memori dan register.  Kelompok instruksi transfer data terdiri dari : MOV,  MVI,  STA,  LDA, STAX,  LDAX,  LHLD,  XCHG,  XTHL,  SHLD, dan LXI. Instruksi-instruksitersebut tidak mengubah isi register flag.

**A. Instruksi MOV (Move)**

* Instruksi **MOV r1, r2** adalah instruksi untuk mentransfer data dari r2(register sumber) ke r1 (register tujuan). Pengertian kata ‘mentransfer´ di sini adalah ‘menggandakan´ atau ‘mengkopi´, karena pada akhir pelaksanaaninstruksi, isi register tujuan jadi sama dengan isi register sumber.
* Instruksi **MOV r, M** berfungsi untuk mentransfer data dari lokasi memori yang alamatnya berada pada pasangan register HL ke suatu register 8 bit. Alamat memori M ditunjukkan oleh isi pasanganregister HL.
* Instruksi **MOV M,** r merupakan kebalikan dari instruksi MOV r, M.Instruksi MOV M, r ini akan memindahkan data 8 bit yang berada pada suaturegister ke suatu lokasi memori yang alamatnya ditunjukkan oleh pasanganregister HL.

**B. Instruksi  MVI (Move Immediate)**

Instruksi  MVI  r, data 8 bit´ dan instruksi  MVI M, data 8 bit.

* Instruksi MVI M, 8 bit, digunakan untuk mengisi data 8 bit pada suatu lokasi memori yang alamatnya ada pada register HL.

**C. Instruksi STA (Store Accumulator Directly in Memory) dan LDA (Load Accumulator Directly from Memory)**

* Instruksi ‘ STA’ dipergunakan untuk menyimpan data yang berada pada akumulator (register A) ke dalam unit memori pada lokasi alamat (4 digit heksadesimal dari belakang mnemonik STA).
* Instruksi  LDA  digunakan untuk memindahkan data 8 bit dari unit memori lokasi alamat tertentu ke akumulator (register A).

**D. Instruksi LHLD (Load H & L Registers Directly from Memory)**

Berfungsi menyalin data memori pada alamat yang spesifik.

**E. Instruksi SHLD (Store H & L Registers Directly in Memory)**

Berfungsi menyimpan data register H & L langsung dalam memori.

‘X’ dalam nama instruksi transfer data menunjukkan bahwa itu berhubungan dengan sepasang  register (16 bit).

**F. Instruksi STAX (Store Accumulator in Address in Register Pair) dan LDAX (Load Accumulator from Address in Register Pair)**

* Instruksi ‘STAX rp’ digunakan untuk menyimpan isi akumulator kelokasi memori yang alamatnya ditunjukkan oleh suatu pasangan register (rp).
* Instruksi ‘LDAX rp’ berfungsi mengisi akumulator dengan data 8 bit dari lokasi memori yang alamatnya ditunjukkan oleh suatu pasangan register.

**G. Instruksi LXI (Load Register Pair with Immediate data)**

Instruksi LXI rp, d16 ; berfungsi mengisi register pair (rp) dari nilai data d16 (alamat 16 bit).

**H. Instruksi XCHG (Exchange H & L with D & E)**

Berfungsi menukar register H & L dengan register D & E.

**I. Instruksi XTHL ( Exchange Top of Stack with H & L)**

**2. Aritmatika**

Instruksi aritmatika contohnya adalah penjumlahan, penguranagan,penambahan dengan 1 (increment) dan pengurangan dengan 1 (decrement).

Pada bagian ini terdapat berbagai intruksi aritmatika diantaranya : ADD, ADI,ADC, ACI, SUB, SUI, SBB, SBI, INR, DCR, INX, DCX, DAD dan DAA.

**A. Instruksi ADD dan SUB**

* Intruksi ADD dipergunakan untuk melakukan operasi penjumlahan 8 bit.Pada operasi tersebut , isi register A (akumulator) akan ditambahkan dengan isi register 8 bit atau isi lokasi memori yang alamatnya ditunjukan oleh pasangan register HL.
* Intruksi SUB dipergunakan untuk melakukan operasi pengurangan 8bit.Pada operasi tersebut, isi register A (akumulator) akan dikurangi dengan isi register 8 bit, atau isi lokasi memori yang alamatnya ditunjukan oleh pasanganregister HL.

**B. Instruksi ADI dan SUI**

* Instruksi ADI dipergunakan untuk menambahkan isi register A (akumulator)dengan data 8 bit secara langsung (immediate data).
* Intruksi SUI dipergunakan dipergunakan untuk mengurangkan isi register A(akumulator) dengan data 8 bit secara langsung (immediate data).

**C. Instruksi ADC dan SBB**

* Instruksi ADC dipergunakan untuk menambah isi register dengan data 8 bitdan mengikutsertakan carry BIT (Cy) dari operasi sebelumnya. Data 8 bit tersebutdapat diambil dari suatu register atau suatu lokasi memori yang alamatnyaditunjukan oleh register HL.
* Instruksi SBB berfungsi untuk mengurangi isi register A dengandata 8 bitdan mengikutsertakan Borrow bit (Cy) dari operasi sebelumnya. Data 8 bittersebut dapat diambil dari suatu register atau suatu lokasi memori yangalametnya ditunjukan oleh register HL.

**D. Instruksi ACI dan SBI**

* Instruksi ACI berfungsi untuk menambah isi register A dengan data 8 bitsecara langsung (immediate data) dan mengikutsertakan carry bit (Cy) darioperasi sebelumnya.
* Instruksi SBI berfungsi untuk mengurangi isi register A dengan data 8 bitlangsung (immediate data) dan mengikutsertakan Borrow bit (Cy) dari operasisebelumnya.

Instruksi ADD, ADI, ADC, SUB, SBI, SUI dan SBB akan mempengaruhi regidter flag (S, Z, AC, P, dan Cy).

**E. Instruksi INR dan INX**

Perintah penambahan 1 sebenarnya  hanya menambahkan isi suatu lokasi (register atau memori) dengan 1. Untuk 8085 ada 2 perintah yaitu INR dan INX. Semua flag kecuali Cy (sisa) dipengaruhi oleh perintah INR dan tidak ada bendera yang dipengaruhi oleh perintah INX.

* Instruksi INR dipergunakan untuk menambah (menaikan)  dengan 1 isi dari suaturegister 8 bit atau lokasi memori yang alamatnya ditunjukan oleh regidter HL.
* INX  dipergunakan untuk menambah dengan 1 isi suatu pasangan register (16 bit register).

**F. Instruksi DCR dan DCX**

* Instruksi DCR dipergunakan untuk mengurangi dengan 1 isi dari suatu register 8 bitatau suatu lokasi memori yang alamatnya ditunjukan oleh register HL.
* Instruksi DCX  dipergunakan untuk mengurangi dengan 1 isi suatu pasangan register (16 bit register).

Instruksi INR dan DCR akan mempengaruhi atau mengubah bit ± bit dari register flag seperti S, Z dan AC tetapi tidak mempengaruhi bit Cy. Instruksi INX danDCX tidak mempengaruhi bit ± bit dari register flag.

**G. Instruksi DAD**

Instruksi DAD dipergunakan untuk menambahkan isi pasangan register HLdengan isi suatu pasangan register pada operasi penjumlahan16 bit.

**H. Instruksi DAA**

Instruksi untuk mengatur bentuk desimal.

**3. Geser**

Instruksi-instruksi geser digunakan untuk menggeser isi register A.Instruksi-instruksi geser hanya mempengaruhi Carry flag (CY).

**A. Instruksi RAR ( Rotate A Right through carry )**

RAR adalah instruksi yang digunakan untuk menggeser isi register A ke kanan melalui carry flag. LSB dari data yang berada pada register A akan mengisi carry flag, sedangkan isi carry flag sebelumnya pindah menjadi MSB.

**B. Instruksi RAL ( Rotate A Left through carry )**

RAL adalah instruksi yang digunakan untuk menggeser isi register A ke kiri melalui carry flag. MSB dari data yng berada pada register A akan mengisi carryflag, sedangkan isi carry flag sebelumnya pindah menjadi LSB.

**C. Instruksi RRC ( Rotate A Right )**

RRC adalah instruksi yang digunakan untuk menggeser isi register A kekanan. LSB akan mengisi MSB dan juga akan mengisi carry flag, sedangkan datase belumnya yang ada di carry flag akan hilang.

**D. Instruksi RLC ( Rotate A Left )**

RLC adalah instruksi yang digunakan untuk menggeser isi register A ke kiriMSB akan mengisi LSB dan juga akan mengisi carry flag, sedangkan datase belumnya yang ada di carry flag akan hilang.

**4. Logika**

**A. Instruksi logika AND**

* ANA M ( And Memory with A )

ANA M adalah instruksi yang digunakan untuk melakukan operasi logika and antara isi register A dengan memori. Kemudian hasil operasi disimpan secara otomatis di register A.

* ANA R ( And Register with A )

ANA R adalah instruksi yang digunakan untuk melakukan operasi logika and antara isi register A dengan isi suatu register lain. Kemudian hasil operasidisimpan secara otomatis di register A.

* ANI ( And Immediate with A )

ANI adalah instruksi yang digunakan untuk melakukan operasi logika andantara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika AND  mempengaruhi isi register flag, yaitu Zero flag (Z), Sign flag (S),  Parity flag (P),  Carry flag selalu dalam kondisi reset (CY = 0) dan Auxiliary Carry selalu dalam kondisi set (AC = 1).

**B. Instruksi OR**

* ORA M ( Or Memory with A )

ORA M adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan memori. Kemudian hasil operasi disimpan secara otomatis di register A.

* ORA R ( Or Register with A )

ORA R adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan isi suatu register lain.  Kemudian hasil operasi disimpan secara otomatis di register A.

* ORI ( Or Immediate with A )

ORI adalah instruksi yang digunakan untuk melakukan operasi logika or antara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika OR mempengaruhi isi register flag, yaitu Zero flag (Z), Signflag (S),  Parity flag (P),  Carry flag selalu dalam kondisi reset (CY = 0) dan Auxiliary Carry selalu dalam kondisi reset (AC = 0).

**C. Instruksi XOR**

* XRA M ( Exclusive Or Memory with A )

XRA M adalah instruksi yang digunakan untuk melakukan operasi logikaexclusive or antara isi register A dengan memori. Kemudian hasil operasidisimpan secara otomatis di register A.

* XRA R ( Exclusive Or Register with A )

XRA R adalah instruksi yang digunakan untuk melakukan operasi logika exclusive or antara isi register A dengan isi suatu register lain. Kemudian hasil operasi disimpan secara otomatis di register A.

* XRI ( Exclusive Or Immediate with A )

XRI adalah instruksi yang digunakan untuk melakukan operasi logika exclusive or antara isi register A dengan konstanta. Kemudian hasil operasi disimpan secara otomatis di register A.

Instruksi logika Exclusive Or mempengaruhi isi register flag, yaitu Zero flag(Z), Sign flag (S), Parity flag (P), Carry flag selalu dalam kondisi reset (CY = 0)dan Auxiliary Carry selalu dalam kondisi reset (AC = 0).

**5. Pencabangan**

Instruksi pencabangan dibagi menjadi dua, yaitu :

**A. Pencabangan bersyarat (conditional)** : JC, JNC, JZ, JNZ, JP,JM, JPE, JPO.

* JC  (J ump if  Carry Set)

Perintah ini digunakan apabila kita ingin menguji carry bit. Jika carry bit berlogika ‘1’, maka program akan melompat ke alamat yang ditulis pada perintah ’JC addr’.

* JNC  (Jump if Carry Not Set )

Perintah ini digunakan apabila kita ingin menguji carry bit. Jika carry bitberlogika ‘0’, maka program akan melompat ke alamat yang ditulis pada perintah ’JNC addr’.

* JM  (J ump if Minus)

Perintah ini digunakan apabila kita ingin menguji sign bit. Program akan melompat pada alamat tertentu apabila sign bit berlogika ‘1’.

* JP  (J ump if plus)

Perintah ini digunakan apabila kita ingin menguji sign bit. Jika sign bit berlogika ‘0’, maka program akan melompat ke alamat yang ditulis setelah perintah JP.

* JZ  (Jump if Zero)

Perintah ini digunakan apabila kita ingin menguji zero bit. Jika zero bi tberlogika ‘1’ maka program akan melompat ke alamat yang ditulis setelah perintah JZ.

* JNZ (J ump if not zero)

Perintah ini digunakan apabila kita ingin menguji zero bit. Jika zero bit berlogika ‘0’ maka program akan melompat ke alamat yang ditulis setelah perintah JNZ.

* JPE (J ump if Even Parity)

Perintah ini digunakan apabila kita ingin menguji parity bit. Jika parity bit berlogika ‘1’, artinya jumlah logika satu dari suatu data hasil operasi yang mempengaruhi register flag berjumlah genap, maka program akan melompat ke alamat yang ditulis setelah perintah JPE.

* JPO (J ump if Parity Odd)

Perintah ini digunakan apabila kita ingin menguji parity bit. Jika parity bit berlogika ‘0’, artinya jumlah logika satu dari suatu data hasil operasi yang mempengaruhi register flag berjumlah ganjil, maka program akan melompat ke alamat yang ditulis setelah perintah JPO.

**B. Pencabangan tidak bersyarat** : JMP (jump) dan CALL.

* JMP  (Jump)

Perintah ini digunakan apabila kita ingin melompat ke alamat tertentu, akan tetapi tanpa melibatkan proses pengujian terhadap bit-bit yang terdapat pada register flag. Oleh karena itu, perintah jump dimasukkan ke dalam perintah pencabangan tak bersyarat.

* CALL

Perintah ini digunakan apabila kita ingin menggunakan suatu subrutin.

**6. IN/OUT dan Control**

Instruksi IN dan OUT berfungsi untuk mengatur lalu lintas dengan peralatan I/O. Dalam instruksi OUT, sumber selalu accumulator. Dalam instruksi IN, tujuan adalah accumulator. Di belakang operasi IN dan OUT kita hanya perlu mencantumkan nomor pintu ( gate number ) saja, yaitu alamatnya peralatan I/O.

Instruksi Machine Control :

* Instruksi SIM

Berfungsi membuat settingan mask interrupt pada mesin prosesor 8085.

* Instruksi RIM

Berfungsi  membaca mask interrupt pada mesin prosesor 8085.

* Instruksi DI

Berfungsi mengnonaktifkan system interrupt pada mesin prosesor 8085.

* Instruksi EI

Berfungsi mengaktifkan system interrupt pada mesin prosesor 8085

* Instruksi HLT

Berfungsi memberhentikan mesin

* Instruksi NOP

Berfungsi tidak ada operasi apapun pada kontrol mesin

**SET INTRUKSI PADA MIKROPROSESOR 8086**

***Instructions Set  pada microprosesor 8086***

***1.***      ***Instructions Set***

Adalah sekumpulan instruksi lengkap yang dapat dimengerti oleh CPU, instruction sets  berupa kode mesin (machine code) dalam bentuk bilangan biner (binary) dan biasanya di- representasi-kan dalam kode/bahasa rakitan (assembly code).

***a.***      ***Element dari instruction***

Instruction terdiri dari beberapa element, yaitu:

  Operation code (Op code)

         Kode perintah operasi .

  Source Operand reference

         Operand penampung nilai yang akan dipros

  Result Operand reference

         Operand penampung nilai hasil proses.

  Next Instruction Reference

         Penghubung ke instruksi berikutnya.

Operand merupakan obyek dari suatu Op code, operand biasanya ditampung pada salah satu tempat penyimpanan berikut:

  Main memory (or virtual memory or cache)

  CPU register

   I/O device

***b.***      ***Representasi dari Op code***

  Representasi dalam bahasa/kode mesin (machine code) tiap instruction memiliki  bentuk/pola bit yang unik (unique bit pattern).

  Representasi symbolic atau kode/bahasa rakitan (assembly code), lebih manusiawi  well, programmers anyway), contohnya:

         ADD

          SUB

         LOAD

  Operands can also be represented in this way

         ADD A,B

***c.***       ***Tipe – tipe Instruction***

  Data processing

  Data storage (main memory)

  Data movement (I/O)

   Program flow control

***d.***      ***Tipe – tipe Operand***

   Addresses

  Numbers

         Integer/floating point

  Characters

         ASCII etc.

  Logical Data

         Bits or flags

***e.***       ***Tipe – tipe Data pada Prosessor Pentium***

Tipe – tipe data yang dapat ditampung pada prosessor pentium adalah tipe data dengan besar kapasitas tampung sebagai berikut:

  8 bit (Byte)

  16 bit (word)

  32 bit (double word)

   64 bit (quad word)

Keterangan:

Tiap data dialamatkan (Addressing) per- 8 bit untuk tiap unit dengan demikian  untuk yang tipe 32 bit (double word) akan di bagi menjadi 4 unit

***f.***        ***Tipe – tipe Operasi***

Berikut adalah tipe – tipe operasi yang terkait dengan Op code:

  Operasi Data Transfer

Operasi tipe ini melakukan transfer data, pada operasi tranfer data perlu di tentukan:

         Source = sumber data

         Destination = tujuan operasi

         Amount of Data = besaran/nilai data

  Operasi Arithmetic

Operasi tipe ini melakukan operasi – operasi aritmatika sebagai berikut:

         Add

         Subtract

         Multiply

         Divide

  Operasi Logical

Operasi tipe ini melakukan operasi yang bersifat logical sebagai berikut:

         ƒ  AND

         ƒ  OR

         ƒ  NOT

  Operasi Conversion

Operasi tipe ini melakukan operasi konversi bilangan, seperti:

         ƒ  Konversi bilangan biner ke desimal

         ƒ  Konversi bilangan real ke int (bilangan bulat)

  Operasi I/O

Operasi tipe ini merupakan operasi yang berhubungan dengan Input/Output,

contohnya:

         ƒ  Operasi untuk menampilkan suatu nilai ke layar monitor

         ƒ  Operasi untuk mencetak, dsb

   Operasi System Control

Operasi tipe ini biasanya operasi untuk penggunaan sistem operasi

  Operasi Transfer of Control

Operasi tipe ini merupakasn operasi yang mengendalikan flow dari proses,

biasanya berupa percabangan, contohnya:

         ƒ  branch to x if result is zero

         ƒ  ISZ Register1 Æ ISZ artinya: Increment and Skip if Zero

         ƒ  interrupt call

***g.***      ***Mode Pengalamatan (Addressing Mode) untuk Operand***

Terdapat 7 mode pengalamatan operand, yaitu:

  Immediate

   Direct

  Indirect

  Register

  Register Indirect

  Displacement (Indexed)

  Stack

***h.***      ***Mode Immediate Addressing***

Merupakan mode pengalamatan operand secara langsung, pada mode ini operand merupakan bagian dari instruction. Operand merupakan area alamat (address field) dari suatu nilai yang akan diproses

Contoh:           ADD 5

Keterangan: Tambahkan nilai 5 dengan nilai di register accumulator dan simpan hasilnya di                       register accumulator.

***Karakteristik mode immediate:***

         No memory reference to fetch data Æ tidak memakai referensi memory untuk mengambil data

         Fast Æ cepat

         Limited range Æ terbatas dalam jangkauan nilai

Diagram untuk mode immediate addressing:

**1.**      **Mode Direct Addressing**

Merupakan mode pengalamatan operand dimana area alamat (address field) berisi alamat dari suatu nilai yang akan diproses.

Effective Address (EA) = Address field (A)

EA = A

Contoh:            ADD A

*Keterangan:*

         Cari di memory pada alamat A untuk operand (*Look in memory at address A for operand*).

         Tambahkan isi yang ada pada alamat A dengan nilai di register accumulator dan simpan hasilnya di register accumulator. (*Add contents of cell A to accumulator*).

*Karakteristik:*

         *Single memory reference to access data* Menggunakan memory untuk mengakses data 􀃆

         *No additional calculations to work out effective address* Tidak memerlukan 􀃆 kalkulasi untuk mendapatkan effective address

         *Limited address space* Address 􀃆 space yang terbatas.

Diagram untuk mode direct addressing:

**2.**      **Mode Indirect Addressing**

Merupakan mode pengalamatan operand dimana area alamat (address field) berisi alamat dari suatu alamat yang akan menunjukkan alamat dari suatu nilai yang akan diproses. (*Memory cell pointed to by address field contains the address of (pointer to) the operand*)

EA = ( A )

Keterangan:

         Cari di memory alamat A, cari alamat yang tertulis pada A untuk operand ( *Look in A, find address (A) and look there for operand* ).

Contoh:                        ADD ( A )

*Keterangan:*

         Tambahkan isi dari cell yang alamatnya ditunjukkan oleh isi yang terdapat pada A dengan nilai yang ada di register accumulator dan simpan hasilnya di register accumulator. (*Add contents of cell pointed to by contents of A to accumulator*).

*Karakteristik:*

         *Large address space*» memerlukan space address yang besar.

         *May be nested, multilevel, cascaded*» Dapat dibuat nested (bersarang), multilevel dan cascade (bertumpuk).

         *Multiple memory accesses to find operand* pengaksesan 􀃆 memory yang multiple untuk mendapatkan operand sehingga mengakibatkan proses mode ini agak lebih lambat

Diagram untuk mode direct addressing:

**3.**      **Mode Register Addressing**

Merupakan mode pengalamatan operand dimana operand yang akan diproses ditampung/disimpan dalam register yang namanya ditulis di area alamat (address field).  (*Operand is held in register named in address field*)

EA = R

*Karakteristik:*

         *Limited number of registers*» terbatas pada jumlah register yang hanya sedikit

         *Very small address field needed* » karena addres yang kecil pada register, maka:

o   *Shorter instructions* » instruksinya lebih pendek

o   *Faster instruction fetch* » pemasukan data lebih cepat

o   *No memory access* » tidak memerlukan akses ke memory

o   *Very fast execution* » eksekusi sangat cepat

o   *Very limited address space* » tetapi space alamat (*address*) sangat terbatas

Diagram untuk mode direct addressing:

**4.**      **Mode Register Indirect Addressing**

Merupakan mode pengalamatan operand dimana area alamat (address field) pada register berisi alamat dari suatu alamat yang akan menunjukkan alamat di memory dari suatu nilai yang akan diproses.  ( *Operand is in memory cell pointed to by contents of register R* )

EA = ( R )

Diagram untuk mode direct addressing:

**Displacement Addressing**

**EA = A + (R)**

Berdasarkan formula di atas, address field menampung 2 nilai, yaitu:

  A sebagai base value

  R sebagai register yang menampung pertukaran sementara (*that holds displacement*)

  Atau sebaliknya

***Relative Addressing***

Merupakan salah satu versi dari pengalamatan untuk pertukaran ( *displacement addressing* )

R = Program counter, PC

EA = A + (PC)

*Keterangan:*

         *Ambil* *operand* dari A dan dari lokasi yang ditunjukkan oleh program counter (PC) (*get operand from A cells from current location pointed to by PC*)

**Micro Operation**

         *Suatu komputer menjalankan suatu program (A computer executes a program)*

         Suatu komputer melakukan siklus proses memasukkan dan mengambil data atau melakukan eksekusi (*Fetch/execute cycle*)

         *Each* cycle has a number of steps

         Aktifitas tersebut di atas disebut micro-operations

         Tiap step pada micro-operations berupa langkah – langkah kecil (*Each step does very little*).

         Micro-operations merupakan operasi atomic dari CPU (*Atomic operation of CPU*)

***Fetch***

***4 Registers***

         **Memory Address Register** (**MAR**)

             o Terkoneksi dengan address bus

             o MAR melakukan spesifikasi address untuk operasi baca atau tulis

         *Memory* **Buffer Register** (**MBR**)

              o Terkoneksi dengan data bus

   o Menyimpan data untuk di tulis atau menyimpan data terakhir yang dibaca (*Holds data to write or* last *data read*)

         **Program Counter** (**PC**)

    o Menyimpan address instruksi berikutnya yang akan diakses (*Holds address of next         instruction to be fetched*)

         **Instruction Register** (**IR**)

     o Menyimpan address instruksi terakhir yang diakses (*Holds last instruction fetched*)

***Fetch* Sequence *(Urutan Fetch)***

         Address dari instruksi berikutnya berada dalam PC

         Address yang ada pada MAR di masukkan ke address bus

         Control Unit memerintahkan perintah membaca (*Control unit issues READ command*)

         Hasil (data dari memory) berada pada data bus

         Data dari data bus di copy ke MBR

         PC ditambah 1 (secara paralel dengan proses fetch dari memory)

         Data (instruksi) dipindahkan dari MBR ke IR

         MBR sekarang kosong dan siap untuk fetch selanjutnya.

***Fetch Sequence Symbolic (Urutan Fetch secara simbol)***

t1: MAR  » (PC)

t2: MBR » (memory)

      PC  »  (PC) +1

t3: IR  » (MBR)

     (tx = time unit/clock cycle)

Or

t1: MAR  » (PC)

t2: MBR  » (memory)

t3: PC  » (PC) +1

      IR  » (MBR)

***Aturan untuk pengelompokan siklus waktu (Rules for Clock Cycle Grouping)***

         Urutan yang baik harus mengikuti aturan berikut (*Proper sequence must be followed*) :

             o MAR » (PC) harus diikuti dengan MBR » (memory)

         Harus menghindari terjadinya konflik (*Conflicts must be avoided*)

            o Tidak boleh melakukan pembacaan dan penulisan terhadap register yang sama dan pada                waktu yg bersamaan (*Must not read & write same register at same time*)

            o Aktifitas MBR » (memory) & aktifitas IR » (MBR) tidak boleh dilakukan pada siklus                 yang bersamaan

         PC harus ditambahkan 1: PC » (PC) +1

            o Menggunakan ALU

            o Mungkin diperlukan additional micro-operations

***Siklus tidak Langsung (Indirect Cycle)***

         MAR » (IRaddress) artinya: Alamat dari IR diisikan ke MAR

         MBR »  (memory) artinya: isi dari memory diisikan ke MBR

         IRaddress »  (MBRaddress) artinya: alamat dari MBR diisikan ke alamat dari IR

         MBR berisi suatu address (*MBR contains an address*)

         Sekarang IR berada pada kondisi yang sama dengan direct addressing (*IR is now in same state as if direct addressing had been used*)

MENCARI SET INTRUKSI RAM ( ANDROID )

**Cek RAM HP Android**

1. Buka menu **“Setting”**.
2. Lalu pilih **“Application”** yang ada dalamnya
3. Setelah itu klik saja **“Running”**
4. Kemudian akan muncul jumlah RAM hp android yang biasanya ditandai dengan **warna biru** untuk RAM yang sudah digunakan dan **warna abu-abu** yang belum digunakan.

Berbagai macam jenis RAM yang ada di Andoroid juga ada beberapa macam, ada **LPDDR2, LPDDR3 dan LPDDR4** yang biasanya digunakan pada Sony Xperia, Asus Zenfone 5,  Samsung Galaxy S4, Xiaomi Mi Note Pro dan yang lainya. Jika ingin memaksimalkan kinerja dari hp Android itu sendiri, usahakan untuk membeli hp yang memiliki kapasitas RAM 1GB. Hal ini dilakukan agar proses browsing dan loading tidak lemot saat digunakan. Akan tetapi jika sudah terlanjur membeli dengan kapasitas 512MB usahakan untuk melakukan pembersihan data penjelajahan atau chache secara berkala agar memori tidak cepat habis.

Setelah melihat beberapa cara cek RAM Android diatas, maka tidak ada salahnya untuk melakukan pengecekan jumlah ruang yang tersisa. Selain itu kami saranka, untuk tidak menggunakan RAM yang melebihi jumlah pemakaian karena bisa mengganggu software yang lainnya.

**Metode 1: ROEHSOFT RAM Expander $ 9,51 (Root)**  
  
Pada $ 9,51, ROEHSOFT RAM Expander adalah aplikasi yang relatif mahal untuk meningkatkan RAM untuk ponsel Android. Namun, jika perangkat Anda kompatibel, ini adalah salah satu aplikasi yang paling dapat diandalkan untuk manajemen RAM efektif pada Android. Aplikasi ini memiliki beberapa fitur tetapi Anda perlu akses root untuk menggunakan aplikasi.

Langkah 1  
Pertama, Anda perlu memeriksa apakah ROEHSOFT Ram Manager mendukung perangkat Anda karena ini adalah aplikasi berbayar. Anda perlu men-download dan menginstal MemoryInfo dan Swapfile pengujian cek aplikasi untuk memeriksa apakah ponsel Anda kompatibel. Anda dapat men-download aplikasi menggunakan tombol Google Play bawah.Setelah Anda menginstal aplikasi, memulai dan tekan tombol Start RAMEXPANDER Uji sini. Setelah itu, gunakan telepon Anda normal selama sekitar 5 menit. Setelah 5 menit, aplikasi akan memberikan hasil memori. Jika perangkat Anda berlari dengan baik selama mereka 5 menit dan Anda tidak mendapatkan kesalahan, maka Anda baik untuk pergi.

Langkah 2  
Berikutnya, download dan menginstal aplikasi ROEHSOFT Ram Expander pada perangkat Android Anda. Anda bisa mendapatkan aplikasi menggunakan tombol Google Play bawah.

Langkah 3  
Meluncurkan aplikasi dan pilih bahasa yang Anda inginkan. Sebuah popup akan meminta Anda untuk Hibah Akses Root. Cukup klik pada Ok untuk memberikan akses aplikasi root. Setelah Anda dibawa ke homescreen, Anda akan melihat banyak informasi tentang perangkat Anda, termasuk RAM dan memori kartu SD. Anda dapat memeriksa kembali dengan info ini dan melihat apakah proses bekerja.

Langkah 4  
Sekarang, memeriksa Autorun untuk memulai aplikasi boot.

Langkah 5  
Selanjutnya, Anda dapat memilih nilai RAM optimal dengan mengklik tombol Nilai Optimal. Anda juga dapat menggunakan slider swapfile manual untuk meningkatkan penyimpanan kartu SD yang ingin Anda gunakan sebagai RAM tambahan.

Langkah 6  
Sekarang, klik pada Swap aktif. Ini akan memulai RAM proses mengoptimalkan untuk meningkatkan RAM di Android pada perangkat ponsel Anda. Anda mungkin tidak melihat peningkatan RAM di menu pengaturan telepon Anda, tetapi perangkat Anda harus merasa lebih cepat dan lancar. Dalam rangka untuk memeriksa kinerja setelah meningkatkan RAM, Anda dapat memainkan permainan HD yang Anda tidak bisa sebelum karena kinerja yang buruk dan harus berjalan jauh lebih baik pada perangkat Anda.

**Metode 2: Cerdas RAM Booster (No Akar)**  
Pintar RAM Booster adalah aplikasi pihak ketiga sangat ringan yang melakukan pekerjaan yang baik untuk meningkatkan RAM pada ponsel Android. Hal ini sangat mudah digunakan dan Anda dapat meningkatkan RAM pada perangkat Anda melalui klik tombol.

Langkah 1  
Download dan menginstal Pintar aplikasi RAM Booster pada perangkat Android Anda. Anda bisa mendapatkan aplikasi melalui tombol Google Play bawah.

Langkah 2  
Setelah terinstal, buka aplikasi dan klik pada pengaturan. Di sini Anda dapat memilih tingkat dorongan dan menambahkan aplikasi ke daftar putih. Memilih tingkat dorongan kuat secara signifikan akan meningkatkan RAM Anda, tetapi banyak aplikasi bisa terbunuh selama proses tersebut. whitelist pada dasarnya adalah daftar di mana Anda dapat menambahkan aplikasi yang Anda tidak ingin membunuh ketika Anda memulai proses meningkatkan. Misalnya, jika Anda ingin menggunakan Whatsapp setiap saat, Anda dapat menambahkannya ke dalam daftar dan tidak akan terbunuh ketika Anda menggunakan boost pilihan. Anda juga dapat memilih Boost secara otomatis pilihan yang secara otomatis meningkatkan RAM saat ponsel berjalan lambat atau jika RAM sudah penuh. Ini adalah pilihan yang sempurna untuk pengguna yang tidak ingin secara manual meningkatkan RAM ketika telepon tindakan lambat. Memeriksa opsi Pemberitahuan RAM Booster akan membiarkan Anda meningkatkan RAM menggunakan bar pemberitahuan Anda.

Langkah 3  
Jika Anda ingin meningkatkan RAM di perangkat ponsel Android secara manual, cukup klik pada Boost Sekarang atau menggunakan tombol Meningkatkan di bar pemberitahuan Anda. Ini akan membunuh beberapa proses dan perangkat Anda harus langsung merasa lebih cepat.

**Metode 3: RAM Manager Pro $ 3,95 (Root)**  
  
Ini adalah aplikasi besar lainnya yang membantu dalam meningkatkan RAM untuk perangkat Android. Versi dasar tidak memiliki banyak fitur dan hampir tidak berguna sehingga Anda akan memerlukan versi pro jika Anda ingin meningkatkan RAM Anda. Aplikasi ini muncul dari script RAM mengoptimalkan sukses di forum XDA. Ini memberi Anda berbagai profil RAM yang dapat Anda gunakan tergantung pada apa yang ingin Anda lakukan pada ponsel Anda. Anda memerlukan akses root untuk menggunakan aplikasi ini.

Langkah 1  
Download dan menginstal RAM Manager Pro pada perangkat Android Anda. Ada juga versi gratis yang tersedia tapi itu hampir tidak berguna sehingga Anda perlu untuk mendapatkan versi pro. Anda dapat men-download melalui Google tombol Play bawah.

Langkah 2  
Meluncurkan aplikasi dan klik pada Ukuran VM Heap. Ini pada dasarnya adalah memori maksimum yang sebuah aplikasi dapat digunakan pada ponsel. Jika beberapa aplikasi Anda memiliki kecenderungan untuk menggunakan banyak RAM, maka Anda dapat menetapkan batas menggunakan opsi ini.  
  
Langkah 3  
Anda juga dapat mengaktifkan swap dengan mengklik pada file Swap. Menggunakan ini tidak dianjurkan karena hal ini dapat dengan mudah merusak kartu SD Anda.

Langkah 4  
Selanjutnya, Anda dapat memilih profil yang ingin Anda gunakan untuk meningkatkan RAM di perangkat ponsel Android. Ini adalah hal terbaik tentang aplikasi yang Anda dapat menggunakan berbagai pengaturan tergantung pada apa yang ingin Anda lakukan pada perangkat Anda. Untuk HD game, Anda dapat memilih profil keras gaming dan untuk menggunakan berbagai aplikasi secara bersamaan, Anda dapat menggunakan opsi multitasking keras. Advanced pengguna juga dapat menggunakan pengaturan sendiri lebih suka dengan mengklik profil kustom. Setelah Anda memilih profil, Anda sudah selesai. Telepon Anda sekarang harus mengelola RAM sesuai dengan profil Anda pilih dan manajemen RAM harus jauh lebih efisien sekarang.

**Metode 4: RAM Kontrol Ekstrim (No Akar)**  
  
RAM Kontrol Extreme aplikasi yang layak lain untuk meningkatkan RAM di perangkat ponsel Android. Aplikasi ini tidak memerlukan akses root dan memiliki beberapa fitur built-in, termasuk pembunuh tugas. Aplikasi ini telah menerima lebih dari 100.000 download, namun beberapa fitur premium hanya disertakan dalam versi Pro dari aplikasi, yang biaya sekitar $ 2,99.

Langkah 1  
Download dan menginstal RAM Kontrol Ekstrim di perangkat Android Anda. Anda bisa mendapatkan aplikasi dengan mengklik tombol di bawah.  
  
Langkah 2  
Membuka aplikasi, dan pergi ke tab PENGATURAN. Di sini Anda dapat memilih RAM meningkatkan tingkat seperti biasa, kuat atau ekstrim. Anda juga dapat memilih auto RAM meningkatkan frekuensi untuk secara otomatis meningkatkan RAM pada perangkat Android Anda.

Langkah 3  
Selanjutnya, buka tab RAMBOOSTER. Sekarang, klik pada tombol Start. Ini akan memulai penguat Auto RAM dan aplikasi akan mulai meningkat RAM pada perangkat ponsel Anda secara otomatis secara berkala sehingga Anda selalu memiliki RAM bebas cukup untuk multitasking.  
  
Langkah 4  
Dalam rangka meningkatkan RAM di perangkat ponsel Android secara manual, Anda dapat pergi ke tab tugaspembunuh. Di sini, Anda dapat memilih aplikasi yang ingin Anda membunuh. Setelah memilih aplikasi, cukup klik pada tombol membunuh untuk membunuh aplikasi dan meningkatkan RAM

**Metode 5: Meningkatkan RAM Umum (No Root / No Apps)**  
Ini adalah satu set tips daripada metode yang sebenarnya untuk meningkatkan RAM di perangkat ponsel Android. Jika Anda tidak memiliki akses root dan tidak ingin menginstal aplikasi pihak ketiga, maka Anda dapat menggunakan tips ini untuk meningkatkan RAM di Android tanpa root.

Langkah 1  
Jangan menggunakan Tinggal wallpaper jika Anda ingin meningkatkan RAM di perangkat ponsel Android. wallpaper hidup memang terlihat indah, tetapi mereka mengambil banyak memori. Anda akan terkejut pada berapa banyak RAM Anda dapat membebaskan dengan beralih ke wallpaper normal.

Langkah 2  
Hapus aplikasi yang tidak perlu yang telah terinstal dan tidak menggunakan sama sekali. Banyak aplikasi berjalan di latar belakang, mengambil banyak RAM. Jika Anda tidak ingin menghapus aplikasi ini dan hanya ingin mereka dijalankan ketika Anda benar-benar ingin menggunakannya, cukup memaksa mereka ditutup dengan pergi ke Settings / Apps. Hal ini harus meningkatkan RAM di perangkat ponsel Android drastis.

Langkah 3  
Jika Anda memiliki banyak aplikasi bloatware bahwa Anda tidak dapat menghapus, Anda dapat menonaktifkan atau kekuatan menutup mereka dengan pergi ke Settings / Apps.

**Meningkatkan Ruang Android Ponsel RAM**  
  
perangkat Android dengan ruang RAM rendah dan versi usang dari sistem operasi android seperti CupCake, Donut, Ginger Bread dan Honeycomb yang mengarah untuk memperlambat kecepatan prosesor dan berbagai masalah lainnya tidak akan mengganggu Anda lagi. Fakta di arena digital saat ini adalah bahwa kita dapat menemukan di atas ketika kita memiliki gadget yang baru diluncurkan saat di telepon kita menyadari bahwa semakin usang sehingga membuat sebagian besar aplikasi diperbarui berfungsi rumit dan bekerja keluar tidak benar. tech savvy dan Geeks perlu keluar semua tabungan mereka jika mereka memilih untuk gadget mobile dan cerdas baru setiap kali dirilis di pasar untuk memecahkan masalah yang telah disebutkan di atas meskipun solusinya sangat sederhana.  
  
Bermain high end game grafis di gadget android tua cerdas, menyingkirkan menggantung ponsel pintar dan menabrak di gadget pintar Anda hanya mengikuti langkah-langkah yang diilustrasikan sebagai berikut. Semua yang disebutkan di atas isu-isu menaikkan karena kurangnya RAM di gadget pintar sehingga apa yang perlu semua pengguna lakukan adalah bahwa untuk meng-upgrade ruang RAM yaitu, untuk meningkatkan ruang RAM. Berikut beberapa kerja dan bereksperimen metode terbaik untuk meningkatkan ruang RAM telah dikutip menindaklanjuti mereka dan menambah ruang RAM Anda dari perangkat android.

Metode Kerja Terbaik untuk Meningkatkan RAM Perangkat Android  
Menambah ruang RAM Anda dengan dua metode terbaik yang memungkinkan meningkatkan dan meningkatkan ruang RAM dari ponsel pintar atau gadget yang berjalan dengan OS android versi usang. Kedua metode yang benar-benar gratis sedangkan metode awal didasarkan pada menggunakan aplikasi dari Google Play Store dan metode lain memungkinkan pengguna untuk meningkatkan ruang RAM secara manual.

RAM KENAIKAN ANDROID PHONE MENGGUNAKAN APP  
Meningkatkan Android dioperasikan gadget ruang RAM pintar dengan menginstal dan menjalankan aplikasi bernama ROEHSOFT RAM EXPANDER (SWAP).  
  
ROEHSOFT RAM Expander adalah sebuah aplikasi luar biasa yang memungkinkan pengguna untuk pengguna Android dioperasikan gadget pintar untuk membuat partisi sebagai seperti hard disk dari desktop pada kartu SC atau kartu memori sehingga dapat memanfaatkan kartu SD atau kartu memori sebagai RAM diperpanjang . Yang berarti lebih banyak ruang pada kartu SD atau kartu memori begitu luas akan menjadi ruang RAM Anda dan bertindak sebagai RAM virtual .