**TUTORIAL PEMBUATAN PROTOTYPE PENDETEKSI KEBAKARAN (FiDo)**

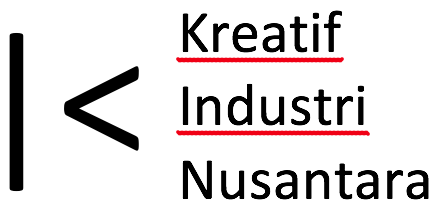
**BERBASIS IoT DENGAN**

**METODE NAÏVE BAYES**

**Mohamad Nurkamal Fauzan, S.T., M.T.**

**Si Made Angga Dwitya P**

Informatics Research Center



**Kreatif Industri Nusantara**

**ABSTRAK**

Kebakaran sering terjadi di daerah perumahan, fasilitas umum, perkantoran ataupun di dalam hutan. Kemunculan kebakaran sendiri sering meninggalkan korban jiwa yang tidak sedikit. Kebakaran terjadi dikarenakan beberapa faktor yaitu faktor manusia yang terjadi karena kelalaian dalam penggunaan bahan kimia atau kurangnya pengawasan. Faktor teknik yang terjadi karena gunung meletus atau sambaran petir. Pada alat-alat pendeteksi kebakaran yang ada di pasar saat ini, menggunakan prinsip kerja yang hanya menggunakan 2 pendeteksi yaitu asap dan Suhu. Tetapi alat – alat tersebut tidak dapat menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.

Oleh karena itu dibuatlah sebuah prototipe pendeteksi kebakaran FiDo ini agar dapat mengetahui kondisi suatu ruangan dalam kondisi aman atau bahaya.Alat ini menggunakan sensor asap, api, dan suhu sebagai pendeteksi dan sebagai data dalam menentukan kondisi suatu ruangan. Alat ini juga dilengkapi dengan fitur notifikasi ke pengguna dan mekanisme penyemprotan air pada saat terdeteksi kebakaran. Penelitian ini menggunakan metode Naïve Bayes untuk kondisi atau status dari suatu ruangan dalam kondisi aman atau bahaya. Metode ini dipilih karena merupakan salah satu metode klasifikasi yang cukup baik dimana kelas penggolongannya telah ditentukan sejak awal. Tujuan dari penelitian adalah untuk membuat alat yang dapat mendeteksi kebakaran dan memberikan informasi mengenai kondisi dari suatu ruangan.Hasil penelitian ini adalah prototype pendeteksi kebakaran FiDo ini diuji menghasilkan tingkat keakurasian sebanyak 89 %.

**Kata Kunci: Kebakaran, Sensor Api, Sensor Asap, Sensor Suhu, Naïve Bayes**

***ABSTRACT***

*Fires often occur in residential areas, public facilities, offices, or the forest. The emergence of fires itself often leaves many casualties. Fires occur due to several factors, namely human factors that occur due to negligence in the use of chemicals or lack of supervision. Engineering factors that occur due to volcanic eruptions or lightning strikes. In fire detection devices on the market today, using a working principle that only uses 2 detectors, namely smoke, and temperature. But these tools cannot determine the condition of a room in a safe or dangerous condition.*

*Therefore a FiDo fire detector prototype was made to know the condition of a room in a safe or dangerous condition. This tool uses smoke, fire, and temperature sensors as detectors and as data in determining the condition of a room. This tool is also equipped with a notification feature to the user and a water spray mechanism when a fire is detected. This study uses the Naïve Bayes method for the condition or status of a room in a safe or dangerous condition. This method was chosen because it is a fairly good classification method in which the classification class has been determined from the start. The purpose of this research is to make a tool that can detect fires and provide information about the conditions of a room. The results of this study are that the FiDo fire detection prototype was tested to produce an accuracy rate of 89%.*

*Keywords: Fires, Flame Sensor, Smoke Sensor Temperature Sensor, Naïve Bayes*

**KATA PENGANTAR**

Puji syukur penulis panjatkan kehadirat Allah SWT karena atas berkat rahmat dan karunia-Nyalah sehingga penulis telah diberikan kekuatan dan kesabaran dalam proses pembuatan dan penyelesaian Buku ini.

Penulis Membuat Buku Yang Berjudul *“Prototype Pendeteksi Kebakaran (Fido) Berbasis Iot Dengan Menggunakan Metode Naive Bayes”.* Penulis juga berharap semoga buku ini tidak hanya sebagai salah satu pemenuhan tugas semata melainkan dapat berguna bagi pembaca .

Selama menyusun buku ini, penulis banyak memperoleh bimbingan, pengarahan, saran, dukungan serta bantuan dari berbagai pihak mulai dari saat mempersiapkan, menyusun, hingga dapat menyelesaikan buku. Maka pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada semua pihak yang turut membantu dalam penyusunan buku yaitu:

1. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Program Studi D-IV Teknik Informatika.
2. Woro Isti Rahayu, S.T., M.T. selaku kordinator
3. Mohamad Nurkamal Fauzan, S.T., M.T. selaku Pembimbing
4. Syafrial Fachrie Pane, S.T., M.T.I., EBDP, selaku dosen pembimbing
5. Nisa Hanum Harani, S.Si., M.T. sebagai dosen wali penulis yang sangat membantu dan memberikan dukungan menyelesaikan buku ini.
6. Seluruh dosen program studi Teknik Informatika yang telah memberikan ilmu pengetahuan dan wawasan yang berguna bagi penulis.
7. Seluruh pengurus perpustakaan, yang telah menyediakan banyak referensi yang tentunya sangat mendukung penulis dalam penyelesaian buku.
8. Orang tua khususnya Ibu serta Kakak Penulis yang telah memberikan dukungan baik moril maupun materil.
9. Teman – teman seperjuangan penulis Teknik Informatika 4B
10. Seluruh pihak yang telah membantu dalam penyelesaian penyusunan laporan ini sesuai yang diharapkan yang tidak bisa penulis sebutkan satu persatu.

Semoga Allah SWT melimpahkan kasih sayang-Nya serta membalas kebaikan semua pihak yang telah membantu dalam penyusunan laporan ini.

Dalam penulisan, penulis menyadari bahwa masih banyak kekurangan dalam buku ini mengingat keterbatasan pengetahuan penulis. Oleh karena itu penulis senantiasa menerima segala kritikan dan saran yang sifatnya membangun dalam penyempurnaan buku ini. Sehingga buku ini dapat menjadi lebih baik dan bermanfaat bagi semua pembaca.

Bandung, 03 September 2020

Penulis

**DAFTAR ISI**

ABSTRAK i

*ABSTRACT* ii

KATA PENGANTAR iii

DAFTAR ISI v

DAFTAR GAMBAR vii

DAFTAR SINGKATAN x

1.1 Latar Belakang I-1

1.2 Identifikasi Masalah I-3

1.3 Tujuan I-4

1.4 Ruang Lingkup I-4

1.5 Sistematika Penulisan I-4

BAB II LANDASAN TEORI II-6

2.1 Teori Umum II-6

2.1.1 Kebakaran II-6

2.1.2 *Internet of Things* II-7

2.2 Pencarian *Source Code*  II-15

2.2.1 GitHub II-15

2.3 Komponen Pada Alat II-16

2.3.1 NodeMCU II-16

2.3.1 Sensor Api II-17

2.3.2 Sensor Asap II-18

2.3.2 Sensor Suhu II-19

2.3.2 Pompa Air II-20

2.4 Metode Yang Digunakan II-21

2.4.1 Metode Naïve Bayes II-21

2.5 Tinjauan Pustaka II-25

BAB III GAMBARAN OBJEK STUDY III-27

3.1 Objek Study III-27

3.2 Sumber Data III-27

3.2.1 Data Primer III-28

BAB IV METODOLOGI PENELITIAN IV-29

4.1 Diagram Alur Metodologi Penelitian IV-29

4.2 Tahapan-Tahapn Diagram Alur Metodologi Penelitian IV-30

4.2.1 Identifikasi Masalah IV-30

4.2.2 Studi Literatur IV-30

4.2.3 Pengumpulan Data IV-30

4.2.4 Pengolahan Data IV-31

4.2.4.1 Metode Naïve Bayes IV-31

4.2.5 Pengujian IV-32

4.2.6 Evaluasi IV-33

BAB V EXPERIMENT DAN RESULT V-34

5.1  *Experiment* V-34

5.1.1 Penerapan Metode Naïve Bayes V-37

5.2 Perancangan V-37

5.2.1 Alur Kerja Sistem V-38

5.2.2 Diagram Blok V-39

5.2.3 Perancangan *Prototype* Pendeteksi Kebakaran (FiDo) V-40

5.2.4 Perancangan *Hardware*  V-41

5.2.5 Perancangan Software V-42

5.2.6 Pembuatan Program V-42

5.3 *Result* V-47

5.3.1 Hasil Penerapan Metode Naïve Bayes Pada Alat V-47

5.3.2 Output Kondisi Pada Saat Tidak Terjadi Kebakaran V-48

5.3.3 Output Kondisi Pada Saat Terjadi Kebakaran V-48

5.3.4 Output Notifikasi pada Telegram V-49

5.3.5 Output Keseluruhan Alat V-51

BAB VI KESIMPULAN VI-51

6.1 Kesimpulan Masalah VI-51

6.2 Kesimpulan Metode VI-51

6.3 Kesimpulan Pengujian VI-51

BAB VII KESIMPULAN VII-52

7.1 Diskusi VII-52

DAFTAR PUSTAKA

LAMPIRAN

w

**DAFTAR GAMBAR**

Gambar 2.1 Konsep *Internet of Thin*gs II-8

Gambar 2.2 Alur kerja pada Github II-10

Gambar 2.2 NodeMCU II-16

Gambar 2.3 Sensor Api II-17

Gambar 2.4 Sensor Asap II-19

Gambar 2.5 Sensor Suhu II-19

Gambar 2.4 Pompa Air II-20

Gambar 4.1 Alur Metodologi Penelitian IV-29

Gambar 5.1 Flowmap Prototipe FiDo V-38

Gambar 5.2 Diagram Blok Prototype FiDo V-39

Gambar 5.3 Skematika pada Prototype FiDo V-40

Gambar 5.4 Interface Arduino IDE V-42

Gambar 5.6 Output Kondisi jika tidak terjadi kebakaran V-48

Gambar 5.7 Output Kondisi jika terjadi kebakaran V-48

Gambar 5.8 Output Notifikasi pada Telegram V-49

Gambar 5.10 Hasil Keseluruhan Alat V-51

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Negara Indonesia merupakan negara dengan populasi terpadat di dunia ke-4 setelah China, India, dan Amerika Serikat. Menurut Badan Perencanaan Pembangunan Nasional, jumlah penduduk Indonesia sendiri pada tahun 2015 sampai dengan 2020 berjumlah sebesar 255.461.700 jiwa dengan laju pertumbuhan penduduk sebesar 1,19.[1] Dengan jumlah penduduk yang semakin meningkat ini mengakibatkan kebutuhan primer berupa papan pada setiap diri individu masyarakat akan semakin meningkat juga dan hal ini dapat menimbulkan sebuah permukiman yang padat penduduk yang membuat rumah antar rumah yang lainnya saling berhimpitan.[2] Salah satu masalah yang dialami pada suatu permukiman yang padat penduduk ini adalah bencana kebakaran

Secara umum Kebakaran merupakan terjadinya reaksi kimia dioksidasi pada bahan bakar yang terjadi sangat cepat dan menghasilkan panas.[3] Kebakaran adalah salah satu bencana yang sering terjadi di masyarakat dan dapat mengakibatkan kerugian yang cukup besar. Kebakaran dapat meluas dan membesar apabila tidak ditangani dengan cepat. Kebakaran terjadi dikarenakan beberapa faktor yaitu faktor manusia yang terjadi karena kelalaian dalam penggunaan bahan kimia atau kurangnya pengawasan.

Faktor Teknik yang terjadi karena gunung meletus atau sambaran petir dan faktor teknis yang terjadi karena reaksi kimia atau listrik. [4] Menurut data dari Badan Nasional Penanggulangan Bencana (BNPB) dalam kurun waktu 7 tahun sejak 2010 hingga 2017 telah terjadi bencana kebakaran sebanyak 1212 kejadian. Jumlah ini mengakibatkan bencana kebakaran menenempati peringkat pertama bencana non alam. [5]

Dengan perkembangan teknologi yang sangat pesat ini, alat-alat pendeteksi kebakaran sudah banyak dijual di pasaran. Pada alat-alat pendeteksi kebakaran yang ada di pasar saat ini, menggunakan prinsip kerja yang hanya menggunakan 2 pendeteksi yaitu suhu dan asap. Selain itu terdapat bunyi peringatan dan mekanisme penyemprotan air dengan penyaluran pipa yang sudah dirancang sebelumnya.[6] Tetapi alat – alat tersebut tidak dapat menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.

Untuk mengatasi permasalahan itu pada penelitian ini dibuatlah *prototype* pendeteksi kebakaran(FiDo) berbasis IoT dengan menggunakan metode Naïve Bayes. Sensor yang digunakan adalah sensor api dan Asap sebagai sensor dalam mendeteksi kebakaran.

*Internet of Things* (IoT) adalah jaringan global yang dinamis infrastruktur dengan kemampuan mengkonfigurasi diri berdasarkan protokol komunikasi standar dan interoperabel di mana "hal-hal" fisik dan virtual memiliki identitas, atribut fisik, dan kepribadian virtual dan menggunakan antarmuka cerdas, dan terintegrasi dengan mulus ke dalam jaringan informasi, sering mengkomunikasikan data yang terkait dengan pengguna dan lingkungannya.[7]

Penerapan IoT pada penelitan ini dengan menggunakan telegram untuk memberi notifikasi kepada pengguna. Sensor Api adalah Sensor yang mempunyai fungsi untuk pendeteksi nyala api, sedangkan sensor asap adalah sensor yang digunakan untuk mengetahui kondisi Asap disekitarnya. Adapun juga sensor suhu yang digunakan untuk mendeteksi suhu pada suatu objek. [8] Sensor api, suhu, dan asap ini berguna untuk mendukung dalam mendeteksi kebakaran.

Metode klasifikasi Naive Bayes adalah salah metode untuk mengklasifikasi dengan menggunakan teknik prediksi peluang kejadian yang sederhana dan mendasar.[9] Dalam penerapannya aturan bayes mengasumsikan bahwa setiap atribut memiliki independensi yang kuat artinya bahwa setiap nilai pada sebuah atribut tidak berkaitan dengan adanya nilai yang sama atau tidaknya dengan atribut lain dalam data yang sama. [10] Pada Penelitan ini metode ini digunakan untuk mengklasifikasikan kondisi atau status dari suatu ruangan dalam kondisi aman atau bahaya

Berdasarkan hasil dari permasalahan diatas, dapat disimpulkan bahwa pada penelitian ini akan memberikan solusi tentang mendeteksi kebakaran dan selanjutnya penggunaan metode naïve bayes digunakan untuk mengklasifikasikan kondisi atau status dari suatu ruangan dalam kondisi aman atau bahaya. Dengan demikian dengan adanya penelitaan ini pendeteksian kebakaran semakin akurat dan kekurangan pada alat-alat sebelumnya bisa diatasi.

* 1. **Identifikasi Masalah**

Dari latar belakang diatas dapat disimpulkan rumusan masalah antara lain:

1. Bagaimana cara mengimplementasikan Metode Naïve Bayes pada prototipe pendeteksi kebakaran FiDo.
2. Bagaimana cara menerapkan IoT pada prototipe pendeteksi kebakaran FiDo
   1. **Tujuan**

Adapun tujuan dari penelitian ini sebagai berikut:

1. Mengetahui cara mengimplementasikan metode Naïve Bayes pada prototype pendeteksi kebakaran FiDo.
2. Menerapkan IoT pada prototipe pendeteksi kebakaran FiDo.

**1.4 Ruang Lingkup**

Berikut ini adalah ruang lingkup masalah dari penelitian yang terdapat pada laporan yang dimana pengujiannya dilakukan di sebuah ruangan pada rumah Kos-Kosan.

1. Terdapat sensor berupa sensor api,sensor asap, sensor suhu untuk mendeteksi nyala api ,suhu dan asap disekitarnya.
2. Notifikasi lewat Telegram
3. Terdapat mekanisme penyemprotan air.

**1.5 Sistematika Penulisan**

Penyajian laporan Tugas Akhir dibagi menjadi beberapa Bab dengan tujuan untuk mempermudah pencarian data atau informasi yang dibutuhkan, serta menunjukan penyelesaian pekerjaan yang sistematis. Pembagian Bab tersebut adalah sebagai berikut:

**BAB I PENDAHULUAN**

Latar Belakang Masalah berisi ulasan ringkas mengenai keadaan atau kondisi yang ada dan kekurangan dari sistem yang diamati sehingga muncul topik yang diambil. Identifikasi Masalah berisi berbagai masalah yang sudah dikenali dan akan diberikan solusinya melalui fungsi dan aplikasi yang akan dibuat. Tujuan berisi tujuan untuk apa aplikasi dibuat. Ruang Lingkup berisi batasan-batasan internship yang akan dibangun. Sistematika Penulisan menjelaskan isi yang ada didalam internship.

**BAB II LANDASAN TEORI**

Membahas tentang konsep dasar dan pengertian yang mendukung terbentuknya “*Prototype* Pendeteksi Kebakaran (*FiDo*) Berbasis IoT dengan menggunakan metode Naïve Bayes”

**BAB III GAMBARAN STUDY CASE**

Pada bab menjelaskan mengenai gambaran dari alat yang akan dibuat.

**BAB IV METODOLOGI PENELITIAN**

Pada bab menjelaskan mengenai tahapan- tahapan alur diagram metodologi penelitian, data yang terkumpul, dan juga pengolahan data yang dilakukan.

**BAB V *EXPERIMENT DAN RESULT***

Membahas mengenai perancangan dan hasil dari alat yang digunakan pada alat pendeteksi kebakaran

**BAB VI *CONCLUSION***

Membahas mengenai hasil dari masalah, metode, dan ekperimen dari alat pendeteksi alat kebakaran

**BAB VII *DISCUSSION***

Membahas mengenai saran yang diberikan kepada pembaca agar dapat mengembangkan alat ini nantinya.

**BAB II**

**LANDASAN TEORI**

* 1. **Teori Umum** 
     1. **Kebakaran**



Kebakaran merupakan terjadinya reaksi kimia dioksidasi pada bahan bakar yang terjadi sangat cepat dan menghasilkan panas. Menurut NFPA (*National Fire Protection Associational*) kebakaran adalah suatu reaksi oksidasi terdiri dari 3 unsur yaitu, bahan bakar, oksigen, dan sumber panas, yang mempunyai dampak kerugian harta benda, cidera, bahkan kematian. Sedangkan menurut departemen tenaga kerja kebakaran merupakan peristiwa yang terjadi karena adanya reaksi oksidasi ekosimetris berlangsung dalam kurun waktu sangat cepat dan disertai dengan timbulnya api.

Kebakaran juga adalah suatu peristiwa yang lebih banyak disebabkan oleh *human error*. kerugian akibat bencana kebakaran antara lain harta benda, terhentinya usaha, bahkan korban jiwa. Kebakaran terjadi dikarenakan beberapa faktor yaitu [11]

1. **Faktor manusia**

yang terjadi karena manusia kelalaian dalam penggunaan bahan kimia atau kurangnya pengawasan.

1. **Faktor Teknik**

yang terjadi karena gunung meletus atau sambaran petir atau gejala alam lainnya yang menyebabkan kebakaran

1. **Faktor Teknis**

yang terjadi karena reaksi kimia atau listrik.

Proses terjadinya kebakaran sendiri dapat berkembang secara cepat ataupun lambat biasanya tergantung berbagai faktor seperti Asap cuaca, arah angin atau berdasarkan material yang terbakar. Kebakaran sendiri juga dapat menimbulkan beberapa kerugiaan yaitu.

1. **Kerugian Kesehatan**

Adalah hal yang paling jelas terlihat. Asap yang ditimbulkan dari kebarakaran dapat menyebabkan berbagai penyakit, terutama infeksi pada saluran pernapasan

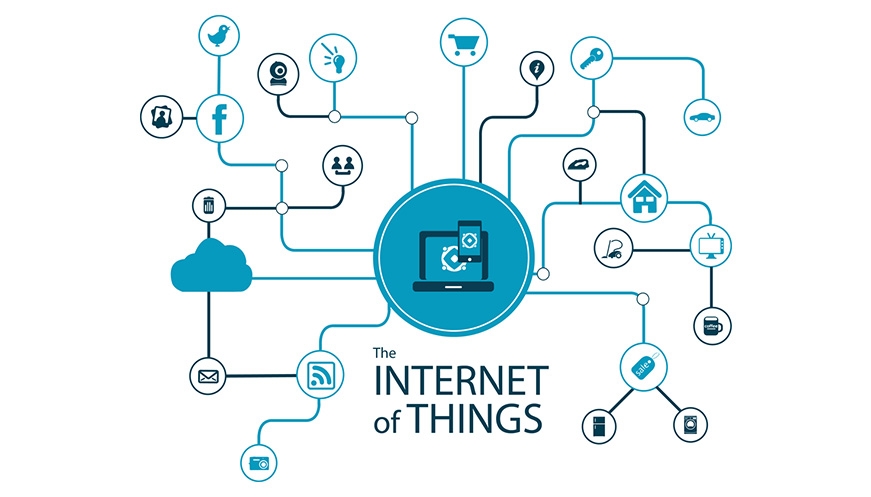
1. **Kerugian Sosial**

Kebakaran dapat menyebakan hilangnya sumber mata pencaharian pekerjaan dan juga korban jiwa

1. **Kerugian Ekonomi**

Kebakaran dapat menyebabkan kerusakan pada fasilitas-fasilitas, serta aset-aset yang dimiliki guna menunjang perekonomian, harta benda yang dimiliki (seperti rumah, uang, perhiasan, dan lain-lain) dan selain juga terdapat biaya kerugian yang harus ditanggung akibat kebakaran tersebut.

* + 1. ***Internet of Things* (IoT)**



IoT adalah jaringan global yang dinamis infrastruktur dengan kemampuan mengkonfigurasi diri berdasarkan protokol komunikasi standar dan interoperabel di mana "hal-hal" fisik dan virtual memiliki identitas, atribut fisik, dan kepribadian virtual dan menggunakan antarmuka cerdas, dan terintegrasi dengan mulus ke dalam jaringan informasi, sering mengkomunikasikan data yang terkait dengan pengguna dan lingkungannya.[12]

*Internet of Things* dalam penerapannya juga dapat mengidentifikasi, menemukan, melacak, memantau objek dan memicu event terkait secara otomatis dan real time, Pengembangan dan penerapan komputer, Internet dan teknologi informasi dan komunikasi lainnya (TIK) membawa dampak yang besar pada masyarakat manajemen ekonomi, operasi produksi, *social* manajemen dan bahkan kehidupan pribadi.

1. **Sejarah Singkat Internet of Things**



Menurut (Burange & Misalkar, 2015) Internet of Things (IOT) adalah struktur di mana objek,orang disediakan dengan identitas eksklusif dan kemampuan untukpindah data melalui jaringan tanpa memerlukan dua arahantara manusia ke manusia yaitu sumber ke tujuanatauinteraksi manusia ke komputer. Internet of Things merupakan perkembangan keilmuan yang sangat menjanjikan untuk mengoptimalkan kehidupan berdasarkan sensor cerdas dan peralatan pintar yang bekerjasama melalui jaringan internet (Keoh, Kumar, & Tschofenig, 2014).Sejak mulai dikenalnya internet pada tahun 1989, mulai banyak hal kegiatan melalui internet, Pada tahun 1990 JohnRomkey menciptakan 'perangkat', pemanggang roti yang bisa dinyalakan dan dimatikan melalui Internet.

WearCam diciptakan pada tahun 1994 oleh Steve Mann. Pada tahun 1997 Paul Saffo memberikan penjelasan singkat pertama tentang sensor dan masa depan.Tahun 1999 Kevin Ashton menciptakan The Internet of Things, direktur eksekutif Auto IDCentre, MIT. Mereka juga menemukan peralatan berbasis RFID (Radio Frequency Identification) global yang sistem identifikasi pada tahun yang sama. Penemuan ini disebut sebagai sebuah lompatan besar dalam commercializing IoT.Tahun 2000 LG mengumumkan rencananya menciptakan kulkas pintar yang akan menentukan sendiri apakah bisa atau tidak makanan yang tersimpan di dalamnya diisi ulang. Pada tahun 2003 RFID mulai ditempatkan pada tingkat besar-besarandi militer AS di ProgramSavi mereka. Pada tahun yangsama melihat raksasa ritelWalmart untuk menyebarkan RFID di semua toko-toko di seluruh dunia untuk lebih besar batas.

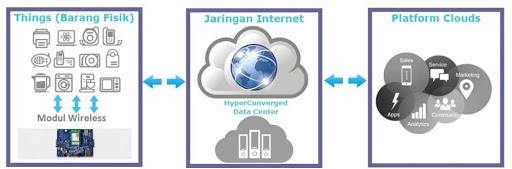
Pada tahun 2005 arus publikasi utama seperti The Guardian,Amerika ilmiah dan Boston Globe mengutip banyak artikel tentang IOT. Pada tahun 2008 kelompok perusahaan meluncurkan IPSO Alliance untuk mempromosikan penggunaan Internet Protocol (IP) dalam jaringan dari "Smart object" dan untuk mengaktifkan Internet of Things.

Pada tahun 2008 FCC menyetujui penggunaan “white space spectrum”. Akhirnya peluncuran IPv6 di tahun 2011 memicu pertumbuhan besar di bidang Internet of Things, perkembangan ini didukung oleh perusahaan raksasa seperti Cisco, IBM, Ericson mengambil inisiatif banyak dari pendidikan dan komersial dengan IOT teknologi dapat hanya dijelaskan sebagai hubungan antara manusia dan komputer.Perkembangan Internet of Things,semua peralatan yang kita gunakan dalam kehidupan kita sehari hari dapat dikendalikan dan dipantau menggunakan IOT. Mayoritas proses dilakukan denganbantuan sensor di IOT. Sensor dikerahkan di mana-mana dan sensor ini mengkonversi data fisik mentah menjadi sinyal digital dan mengirimkan mereka ke pusat kontrol. Dengan cara ini kita bisa memonitor perubahan lingkungan jarak jauh dari setiap bagian dari dunia melalui internet. Arsitektur sistem ini akan didasarkan pada konteks operasi dan proses dalam skenario real-time. Di otomasirumah setiap kotak saklar listrik akan terhubung dengan ponsel pintar (atau kadang-kadang remote) sehingga itu bisa dioperasikan dari jarak jauh. Tapi skenario seperti itu tidak perluprosesor dan perangkat penyimpanan dipasang di setiap kotak saklar. Hanya dibutuhkan sensor untuk menangkap sinyal dan proses itu (kebanyakan beralih ON / OFF). Jadi arsitektur sistem ini bervariasi tergantung pada konteks penerapannya (Suresh et al., 2014).

1. **Konsep dari *Internet of Things***

Konsep IoT ini sebetulnya cukup sederhana dengan cara kerja mengacu pada 3 elemen utama pada arsitektur IoT, seperti:

1. Barang Fisik yang dilengkapi dengan modul IoT
2. Perangkat Koneksi ke Internet berupa Modem dan Router Wireless Speedy seperti di rumah anda.
3. *Cloud Data Center* tempat yang digunakan untuk menyimpan aplikasi beserta data base



*Gambar 2.1 Konsep Internet of Thin*g

1. **Prinsip-Prinsip Internet of Things**

Istilah “*Internet of Things*” terdiri atas dua bagian utama yaitu Internet yang mengatur konektivitas dan Things yang berarti objek atau perangkat. Secara sederhana, kamu memiliki “*Things*” yang memiliki kemampuan untuk mengumpulkan data dan mengirimkannya ke Internet. Data ini dapat diakses oleh “*Things*” lainnya juga. Adapun 7 prinsip dasar menopang IoT:

1. Big Analog Data

Big Analog Data bisa didapatkan dari berbagai macam sumber yang sifatnya alami seperti cahaya, sinyal radio, getaran, suhu, dan sebagainya, serta bisa dihasilkan oleh peralatan mekanis atau elektronik.Big Analog Data adalah tipe Big Data yang terbesar dan tercepat jika dibandingkan dengan tipe-tipe Big Data lainnya. Sehingga, dalam banyak hal, Big Data Analog perlu diperlakukan secara khusus.

1. *Perpetual Connectivity*

Perpetual Connectivity merupakan konektivitas yang terus-menerus menghubungkan perangkat ke Internet. IoT yang selalu terhubung dan aktif dapat memberikan tiga manfaat utama seperti:

1. Monitor: Pemantauan berkelanjutan yang memberikan pengetahuan berisi informasi real time tentang penggunaan suatu produk atau pengguna di lingkungan industri.
2. Maintain: Pemantauan berkelanjutan memungkinkan kita untuk melakukan peningkatan atau tindakan-tindakan tertentu sesuai dengan kebutuhan.
3. Motivate: Konektivitas yang konstan dan berkelanjutan dengan konsumen atau pekerja memungkinkan pelaku usaha atau pemilik organisasi untuk memotivasi orang lain membeli produk, mengambil tindakan, dan sebagainya.
4. *Really Real Time*

Definisi real time untuk IoT berbeda dari definisi real time pada umumnya. Real time sebenarnya dimulai dari sensor atau saat data diperoleh. Real time untuk IoT tidak dimulai ketika data mengenai switch jaringan atau sistem komputer.

1. *The Spectrum of Insight*

“Spectrum of Insight” berasal dari data IoT yang berkaitan dengan posisinya dalam lima fase data flow yaitu real time, in motion (bergerak), early life, at rest (saat istirahat), dan arsip. Masih berhubungan dengan poin sebelumnya tentang real time pada IoT, real time diperlukan untuk menentukan respons langsung dari sistem kontrol. Di ujung lain dari spektrum, data yang diarsipkan di pusat data atau cloud dapat diambil untuk analisis komparatif terhadap data yang lebih baru.

1. *Immediacy Versus Depth*

Dengan berbekal komputer dan solusi IoT di era digital ini, akan ada pertukaran antara kecepatan dan kedalaman yang kita dapatkan. Artinya, seseorang bisa langsung mendapatkan “Time-to-Insight” pada analitik yang belum sempurna seperti perbandingan suhu atau transformasi Fourier cepat untuk menentukan apakah memutar roda pada trem akan menyebabkan kecelakaan. Time (waktu) di sini dibutuhkan untuk mendapatkan insight (wawasan) yang mendalam tentang suatu data. Data yang dikumpulkan membutuhkan waktu yang lama untuk dianalisis dan sejumlah besar perangkat komputer back-end.

1. *Shift Left*

Seperti yang sudah dijelaskan di poin sebelumnya, untuk mendapatkan wawasan yang cepat dan menyeluruh tergolong sangat sulit. Namun, beberapa insinyur berhasil mengatasi kesulitan itu dan mendapatkannya. Fenomena ini disebut dengan “The Genius of the AND”. Drive untuk mendapatkan wawasan tersebut akan menghasilkan komputasi dan analisis data canggih yang biasanya disediakan untuk cloud atau pusat data.

1. *The Next V*

Big Data biasanya ditandai dengan “V” yaitu Volume, Velocity, Variety, dan Value. The next V yang dimaksud adalah Visibility. Ketika data dikumpulkan, para ilmuwan data di seluruh dunia harus bisa melihat dan mengaksesnya sesuai kebutuhan. Visibilitas menawarkan kemudahan yang menjadikan pengguna tidak harus mentransfer sejumlah besar data ke orang atau lokasi yang jauh.

1. **Manfaat dari Internet of Things**

Beberapa manfaat IoT mungkin tidak terlalu kentara, tetapi bukan berarti tidak bisa dirasakan. Di bawah ini adalah tiga manfaat utama yang akan kamu dapatkan langsung dari IoT:

1. Konektivitas

Di era digital ini, kamu bisa mengucapkan selamat tinggal pada era pengoperasian perangkat secara manual. Dengan IoT, kamu bisa mengoperasikan banyak hal dari satu perangkat, misalnya smartphone.

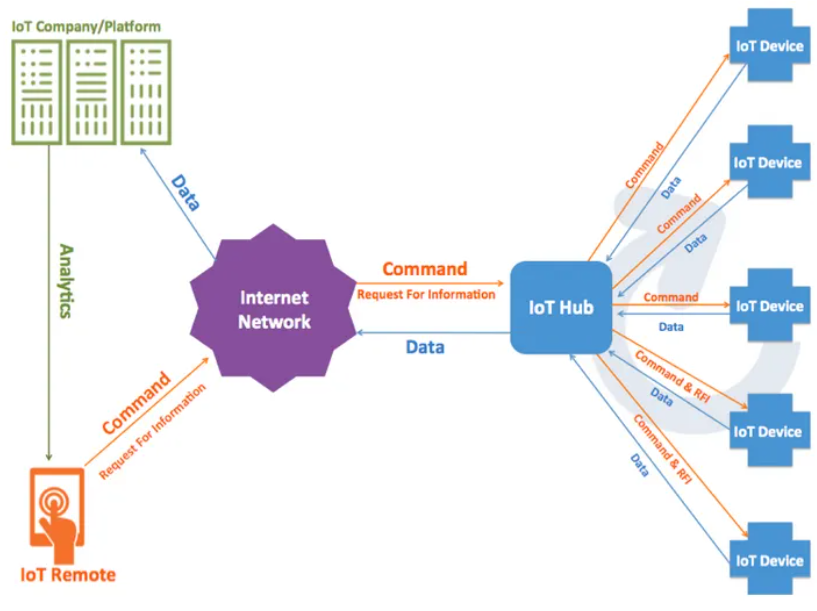
1. Efisiensi

Dengan adanya peningkatan pada konektivitas, berarti terdapat penurunan jumlah waktu yang biasanya dihabiskan untuk melakukan tugas yang sama. Misalnya, asisten suara seperti Apple's Homepod atau Amazon's Alexa dapat memberikan jawaban atas pertanyaan tanpa kamu perlu mengangkat telepon atau menghidupkan komputer.

1. Kemudahan

Perangkat IoT seperti smartphone kini mulai menjadi perangkat yang biasa dimiliki oleh sebagian besar orang. Misalnya smart refrigerator dan Amazon Dash Button yang memudahkan kamu untuk menyusun ulang item dengan hanya satu atau dua tindakan yang menunjukkan persetujuan kamu.

1. **Cara Kerja IoT**

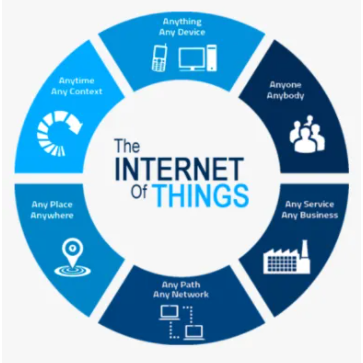


Cara kerjanya adalah dengan memanfaatkan instruksi pada suatu pemrograman, yang masing-masing perintahnya bisa menghasilkan interaksi antar perangkat yang telah saling terhubung. Di sini, anda juga perlu mempelajari Apa Itu Artificial Intelligence.

Cara kerja tersebut sifatnya otomatis dan tidak harus menggunakan campur tangan dari manusia. Internet bisa menjadi penghubung kedua interaksi pada suatu perangkat jarak jauh. Manusia hanya menjadi pengatur dan pengawas ketika alat tersebut sedang bekerja. Penyusunan jaringan komunikasinya yang terbilang cukup kompleks menjadi tantangan paling besar dalam internet of things, sehingga membutuhkan sistem keamanan yang cukup ketat.

Ada banyak manfaat yang bisa kita peroleh dalam kehidupan sehari-hari. Seperti yang telah dibahas tadi, yaitu jaringan yang telah terbentuk oleh benda yang memiliki identitas. Fungsi utamanya adalah untuk memudahkan seluruh kegiatan sehari-hari.

1. **Unsur-Unsur pada Internet of Things**



Ada beberapa unsur di dalam internet of things yang harus diketahui. Unsur tersebut menjadi bagian dari pembuatan IoT itu sendiri, dan perangkat yang ada di dalamnya akan memengaruhi cara kerja internet of thing tersebut. Unsurnya adalah sebagai berikut:

1. Kecerdasan Buatan (*Artificial Intelligence*/AI)

IoT dapat membuat semua mesin yang ada menjadi canggih. Jadi IoT dapat meningkatkan segala aspek yang ada di kehidupan dengan pengembangan teknologi berdasarkan pada AI

1. Konektivitas

Di IoT, ada kemungkinan untuk membuat/membuka jaringan baru, dan jaringan khusus IoT. Jadi, jaringan ini tak lagi terikat hanya dengan penyedia utamanya saja.

1. Sensor

Yang menjadi pembeda dan membuat IoT unik dibanding mesin canggih lainnya. Sensor ini mampu mengartikan segala instrumen, yang mengubah IoT dari jaringan standar dan cenderung pasif dalam perangkat, hingga menjadi suatu sistem aktif yang sanggup diintegrasikan ke dunia nyata sehari-hari kita.

1. Keterlibatan Aktif

Yang sering diterapkan teknologi umumnya yang termasuk pasif. IoT menjelaskan paradigma yang baru bagi konten yang masih aktif, produk, maupun keterlibatan layanan.

1. Perangkat Berukuran Kecil

Perangkat, seperti yang diperkirakan para pakar, memang menjadi semakin kecil, makin murah, dan lebih kuat dari masa ke masa. IoT bisa menggunakan perangkat-perangkat kecil yang dibuat khusus ini agar menciptakan ketepatan, skalabilitas, dan fleksibilitas yang baik.

1. **Hal-Hal Yang Perlu Dipahami Dalam Penggunaan Internet of Things**

Adapun beberapa hal yang perlu dipahami dalam belajar Internet of Things:

1. **Memiliki Pemahaman Lebih Tentang Sensor**

Tidak seperti developer pada umumnya, kamu yang tertarik dengan pemrograman di ruang IoT harus memiliki pemahaman yang mendalam tentang sensor dan komunikasi nirkabel. Selain komputasi, IoT akan membawa kamu ke dunia teknik mesin dan sipil saat sensor mengumpulkan data fisik. Untuk itu, akan lebih baik lagi jika developer IoT memiliki latar belakang ilmu komputer dan teknik.

1. **Belajar JavaScript atau Phyton**

Kamu perlu menguasai bahasa berbasis web untuk back-end pemrosesan data sebelum mengejar karir sebagai developer IoT. JavaScript adalah bahasa yang ideal untuk memproses data baru dari perangkat dan memicu tindakan pada perangkat itu sendiri. Selain itu, Python juga dianggap sebagai bahasa pemrograman umum yang wajib dikuasai terlebih dahulu sebelum memasuki dunia IoT, dibarengi dengan beberapa perangkat yang kompatibel dengan Windows IoT.

1. **Macam-macam Bidang Penerapan IoT**
2. **Pertanian**



Ada berbagai macam pengaplikasani IoT di sektor pertanian. Beberapa diantaranya seperti mengumpulkan data soal suhu, curah hujan, kelembaban, kecepatan angin, serangan hama, dan muatan tanah. Data-data tersebut bisa dipakai buat mengotomatisasi teknik pertanian. Kemudian, bisa juga dipakai untuk mengambil keputusan (decision making) berdasarkan informasi yang ada demi meningkatkan kualitas dan kuantitas, meminimalkan risiko dan limbah, serta mengurangi upaya yang diperlukan dalam mengelola tanaman. Sebagai contoh, petani sekarang sudah bisa memantau suhu dan kelembaban tanah dari jauh, dan bahkan menerapkan data yang diperoleh IoT untuk program pemupukan yang lebih presisi.

1. **Energi**



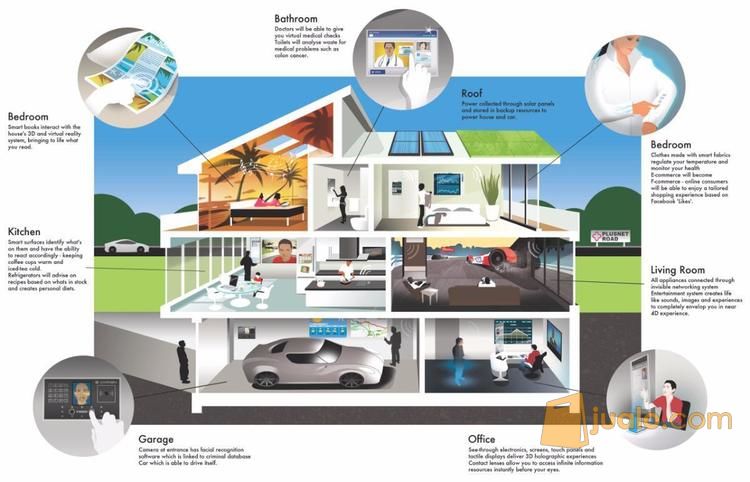
Sejumlah besar perangkat yang memakan energi (semacam switch, outlet listrik, lampu, televisi, dll.) kini sudah bisa terintegrasi dengan konektivitas internet. Lalu apa dampaknya? Integrasi itu memungkinkan mesin-mesin ataupun jaringan untuk berkomunikasi dalam menyeimbangkan pembangkitan listrik serta penggunaan energi yang lebih hemat maupun efektif. Perangkat ini juga bisa memungkinkan akses remote control dari pengguna, atau bisa juga manajemen dari satu pusat lewat interface yang berberbasis cloud. Selain itu, bisa juga mengaktifkan fungsi semacam penjadwalan (misalnya untuk menyalakan/mematikan mesin pemanas, mengendalikan oven, mengubah kondisi pencahayaan dari terang menjadi redup hingga ke gelap, dan lain sebagainya). Jadi dengan IoT di bidang ini, sistem bisa berkumpul dan bertindak berdasarkan informasi yang terkait dengan energi dan daya demi meningkatkan efisiensi produksi dan distribusi listrik.

1. **Lingkungan**



Aplikasi pemantauan lingkungan dari IOT biasanya pakai sensor dalam membantu terwujunya perlindungan lingkungan. Contohnya seperti apa? Penerapannya misalnya dengan memantau kualitas udara atau air, kondisi atmosfer atau tanah, bahkan juga bisa mencakup pemantauan teerhadap satwa liar dan habitatnya. Tak hanya itu sebenarnya. Bisa juga IoT ini dimanfaatkan dalam penanggulangan bencana semacam sistem peringatan dini Tsunami atau gempa bumi. Hal ini tentunya bisa sangat membantu. Perangkat IoT dalam hal ini berarti punya jangkauan geografis yang sangat luas serta mampu bergerak.

1. **Otomatisasi Rumah**



Perangkat IoT juga bisa dipakai untuk memantau dan mengontrol sistem mekanis, elektrik, dan elektronik yang digunakan di berbagai jenis bangunan (misalnya, industri atau juga rumah Anda sebagai tempat tinggal). Alat atau pengembangan IoT ini juga bisa memantau penggunaan energi secara real-time untuk mengurangi konsumsi energi. Tak hanya itu, bahkan bisa juga melakukan pemantauan terhadap para penghuninya. Contohnya? Begitu Anda masuk ke rumah di malam hari, lampu menyala. Kemudian begitu Anda masuk ke jadwal tidur, lampu akan mati secara otomatis. Pagi hari, taman Anda akan disiram air oleh mesin penyiram otomatis. Begitu juga dengan kulkas Anda yang bisa memesan stok makanan sendiri ketika habis. Semuanya bisa terintegrasi menjadi sistem rumah pintar.

* + - 1. **Kegunaan IoT pada Prototipe FiDo**

IoT pada prototipe FiDo untuk pendeteksi kebakaran berfungsi sebagai pemberi informasi kepada pengguna. Informasi yang didapat berisi kondisi atau status dari suatu ruangan dan informasi tersebut dikirimkan melalui aplikasi telegram.

* 1. **Pencarian *Source Code*** 
     1. **Pengenalan Git**



Git adalah version control system yang digunakan para developer untuk mengembangkan software secara bersama-bersama. Fungsi utama git yaitu mengatur versi dari source code program anda dengan mengasih tanda baris dan code mana yang ditambah atau diganti.

Git ini sebenernya memudahkan programmer untuk mengetahui perubahan source codenya daripada harus membuat file baru seperti Program.java, ProgramRevisi.Java , ProgramRevisi2.java, ProgramFix.java. Selain itu, dengan git kita tak perlu khawatir code yang kita kerjakan bentrok, karena setiap developer bias membuat branch sebagai workspacenya.Fitur yang tak kalah keren lagi, pada git kita bisa memberi komentar pada source code yang telah ditambah/diubah, hal ini mempermudah developer lain untuk tahu kendala apa yang dialami developer lain.

**Sejarah Git**

Git diawali dengan sedikit permasalahan dan kontroversi. Kernel Linux merupakan sebuah proyek perangkat lunak open source skala besar. Sepanjang perjalanan perawatan Kernel Linux (1991-2002), perubahan disimpan sebagai patch dan arsip-arsip berkas. Pada tahun 2002, proyek ini mulai menggunakan sebuah DVCS proprietary bernama BitKeeper.

Pada tahun 2005, hubungan antara komunitas pengembang Kernel Linux dengan perusahan yang mengembangkan Bitkeeper retak, dan status “gratis” pada BitKeeper dicabut. Hal ini membuat komunitas pengembang Kernel Linux (dan khususnya Linus Torvalds, sang pencipta Linux) harus mengembangkan perkakas sendiri dengan berbekal pengalaman yang mereka peroleh ketika menggunakan BitKeeper.

Dan sistem tersebut diharapkan dapat memenuhi beberapa hal berikut:

* + 1. Kecepatan
    2. Desain yang sederhana
    3. Dukungan penuh untuk pengembangan non-linear (ribuan cabang paralel
    4. Terdistribusi secara penuh
    5. Mampu menangani proyek besar seperti Kernel Linux secara efisien (dalam kecepatan dan ukuran data)

Sejak kelahirannya pada tahun 2005, Git telah berkembang dan semakin mudah digunakan serta hingga saat ini masih mempertahankan kualitasnya tersebut. Git luar biasa cepat, sangat efisien dalam proyek besar, dan memiliki sistem pencabangan yang luar biasa untuk pengembangan non-linear**.**

Contoh dari software version control system adalah github, bitbucket, snowy evening, dan masih banyak lagi. Jika anda sebagai developer belum mengetahui fitur git ini, maka anda wajib mencoba dan memakainya. Karena banyak manfaat yang akan didapat dengan git ini.

**Fitur-Fitur GIT**

Setelah mengetahui apa itu GIT, selanjutnya anda juga harus mengetahui apa saja fitur fitur yang ada didalam GIT berikut ini :

1. Version control system yang terdistribusi, GIT menggunakan pendekatan peer to peer, tidak seperti yang lainnya seperti Subversion (SVN) yang menggunakan model client-serve
2. GIT memungkinkan developer untuk memiliki branch kode yang independent dan massive. Membuat, menghapus dan menggabungkan branch tersebut menjadi lebih cepat, lancar dan tidak butuh waktu yang lama.
3. GIT bersifat atomic, adalah sebuah tindakan akan benar-benar diselesaikan dengan lengkap atau sama sekali gagal. Ini sangat penting, karena di beberapa version control system seperti CVS, operasinya bersifat non-atomic. Jika ada operasi yang ‘gantung’ dan terkait dengan repository, kondisi repository menjadi tidak stabil.
4. Media penyimpanan GIT yaitu dalam folder.git. Berbeda dengan VCS lain seperti SVN atau CVS, dimana metadata file disimpan dalam folder tersembunyi seperti .cvs, .svn, etc.
5. GIT memiliki data model yang dapat membantu memastikan integritas cryptographic yang ada dalam repository. Sehingga setiap kali sebuah file ditambahkan atau di-commit, checksum-nya akan diciptakan; sama hal nya, juga di-retrieve lewat checksum-nya juga.
6. GIT memiliki staging area atau index. Dengan stagin area, developer bisa memformat commit dan membuatnya bisa di-review sebelum benar-benar diterapkan.
7. GIT sangat sederhana dalam penggunannya. Untuk memulai, Anda bisa membuat repository atau men-checkout yang sudah ada. Setelah instalasi, perintah git-init akan men-setup semuanya.

**Perintah – Perintah Dasar GIT**

Untuk mengetahui bagaimana menggunakan git, berikut perintah-perintah dasar git:

1. **Git init** : untuk membuat repository pada file lokal yang nantinya ada folder .git
2. **Git status** : untuk mengetahui status dari repository lokal
3. **Git add** : menambahkan file baru pada repository yang dipilih
4. **Git commit** : untuk menyimpan perubahan yang dilakukan, tetapi tidak ada perubahan pada remote repository.
5. **Git push** : untuk mengirimkan perubahan file setelah di commit ke remote repository.
6. **Git branch** : melihat seluruh branch yang ada pada repository
7. **Git checkout** : menukar branch yang aktif dengan branchyang dipilih
8. **GIt merge** : untuk menggabungkan branch yang aktif dan branch yang dipilih
9. **Git clone** : membuat Salinan repository lokal

**Keunggulan dan Kelemahan GIT**

**Keunggulan dari Git**

* Design yang mudah dipahami
* Mendukung dalam pengembangan non-linear(pengembanganparalel)
* Sistem terdistribusi, peer to peer
* Mendukung dalam proyek besar seperti Kernel Linux
* Akses hanya menggunakan command line tanpa memerlukan client server
* Penyimpanan murni berbasis file (tidak menggunakan database/SQL)
* Open Source alias Gratis

**2. Kelemahan GIT**

* Tidak Optimal untuk pengembang tunggal
* Dukungan untuk Windows terbatas dibandingkan Linux
  + 1. **GitHub**



GitHub adalah perangkat lunak berbasis web yang dapat digunakan sebagai media untuk berkolaborasi dalam pengembangan proyek perangkat lunak aplikasi. Github adalah manajemen kode sumber (SCM). Model Pembelajaran Kolaboratif dari rekayasa perangkat lunak menggunakan Github untuk mahasiswa informatika tidak hanya memfasilitasi proses pembelajaran tetapi juga memfasilitasi siswa untuk berkolaborasi sehingga tujuan pembelajaran dapat tercapai. [13].

* + - 1. **Sejarah GitHub**

Pada tanggal 24 Februari 2009, anggota tim GitHub mengumumkan, dalam sebuah pembicaraan di markas besar Yahoo!, bahwa saat tahun pertama online, GitHub telah mengumpulkan lebih dari 46.000 repositori publik, 17.000 diantaranya dibuat pada bulan sebelumnya. Saat itu, sekitar 6.200 repositori telah bercabang dan 4600 telah digabung.

Pada tanggal 5 Juli 2009, GitHub mengumumkan bahwa situs itu kini dimanfaatkan oleh lebih dari 100.000 pengguna. Pada tanggal 27 Juli 2009, Dalam pembicaraan lain yang disampaikan di Yahoo!, Tom Preston-Werner mengumumkan bahwa GitHub telah tumbuh menjadi tuan rumah 90.000 repositori publik yang unik, 12.000 yang telah bercabang, untuk total dari 135.000 repositori.

Pada tanggal 25 Juli 2010, GitHub mengumumkan bahwa situs tersebut mempunyai 1 juta repositori. Pada tanggal 20 April 2011, GitHub mengumumkan bahwa situs tersebut mempunyai 2 juta repositori. Pada tanggal 2 Juni 2011, ReadWriteWeb melaporkan bahwa GitHub telah melampaui SourceForge dan Google Code di total nomor dari komit untuk jangka waktu Januari sampai Mei 2011.

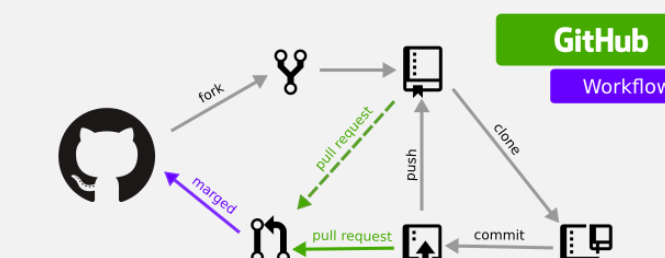
Pada tanggal 9 Juli 2012, Peter Levine, Partner umum di GitHub Investor Andreessen Horowitz, menetapkan bahwa GitHub telah meningkatan pendapatannya pada angka 300% setiap tahun sejak tahun 2008 “menguntungkan hampir seluruhnya”.

Pada tanggal 16 Januari 2013, GitHub mengumumkan telah melewati 3 juta pengguna dan kemudian mempunyai lebih dari 5 juta repositori. Pada tanggal 23 Desember 2013, GitHub mengumumkan telah mencapai 10 juta repositori. Pada bulan Juni 2015, GitHub membuka sebuah kantor di Jepang yang mana itu adalah kantor pertama GitHub di luar Amerika Serikat.

Pada tanggal 29 Juli 2015, GitHub mengumumkan telah mendapatkan dana sebesar 250 juta dollar oleh Sequoia Capital. Nilai perusahaan pada saat putaran ini kira-kira 2 milyar dollar. Pada tahun 2016, GitHub menduduki posisi 14 di Forbes Cloud 100 list. Dengan rilis pertama pada tanggal 21 Juli 2017, browser web Brave menjadikan GitHub sebagai salah satu mesin pencari defaultnya.

Pada tanggal 28 Februari 2018, GitHub menjadi korban serangan DoS terbesar sepanjang sejarah, dengan lalu lintas masuk mencapai sekitar 1,35 terabit per detik. Pada tanggal 4 Juni 2018, Microsoft mengakuisisi GitHub sebesar 7,5 miliar dollar. Pada tanggal 19 Juni 2018, GitHub memperluas GitHub Education dengan menawarkan bundel pendidikan gratis ke semua sekolah**.**

**Fitur -Fitur yang ada pada GITHUB**



*Gambar 2.2 Alur kerja pada Github*

Ada beberapa fitur yang terdapat di GitHub:

1. Repositori atau repo adalah direktori penyimpanan file proyek. Di sini,

bisa menyimpan apa pun yang berkaitan dengan proyek yang sedang

dibuat, misalnya file kode, gambar, atau audio. Repo sendiri bertempat dipenyimpanan atau storage GitHub atau repositori lokal di komputer atau laptop pribadi.

1. *Branch* merupakan hasil salinan dari repositori milik pribadi. Branch

digunakan ketika hendak melakukan suatu pengembangan atau

*development* secara terpisah.

1. *Pull request* adalah ketika Anda menginformasikan user bahwa sudah push

perubahan yang dilakukan di branch ke master repositori. Collaborator

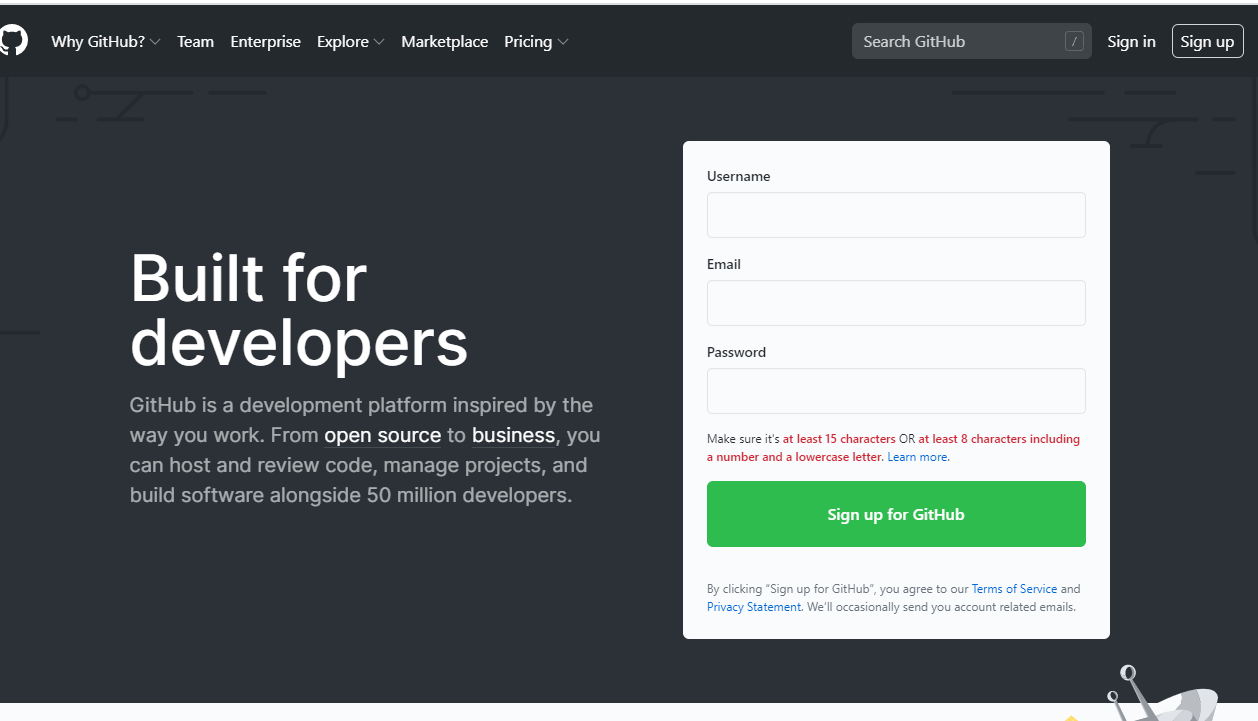
repositori akan menerima atau menolak pull request.

1. *Forking* repositori artinya Anda membuat proyek baru berdasarkan

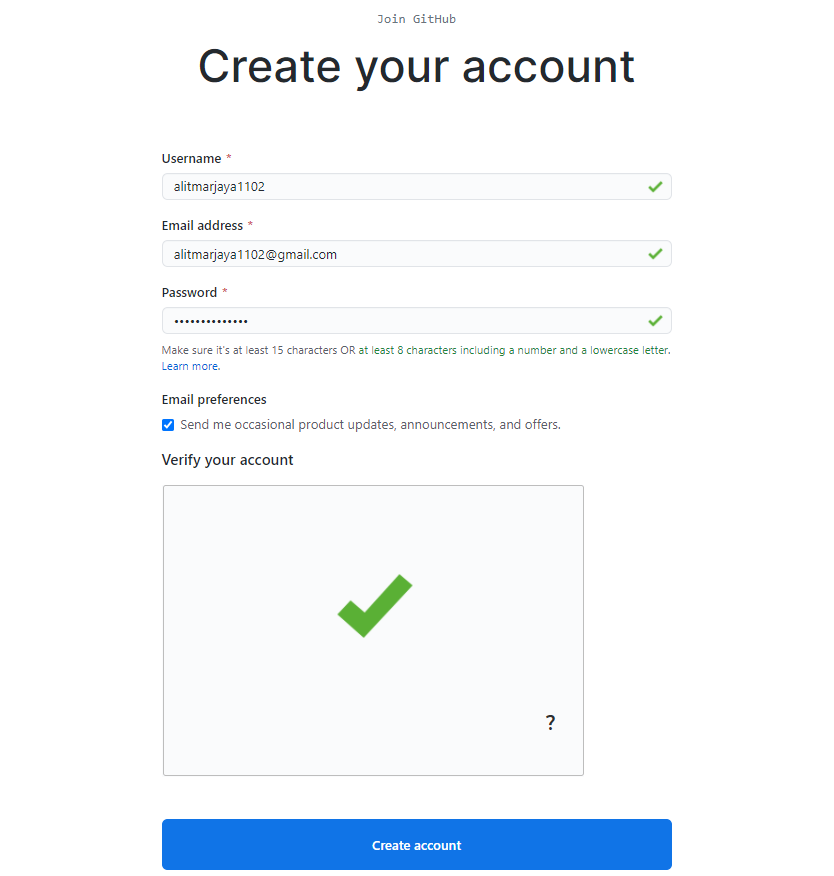
repositori yang sudah ada.

**Cara Penggunaan GITHUB**

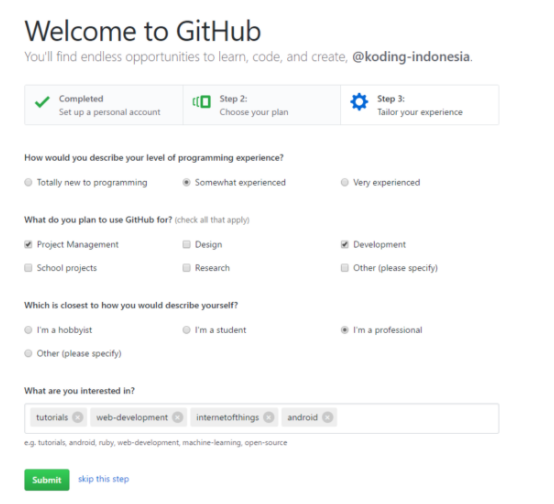
1. Buatlah akun terlebih dahulu di [**http://github.com**](http://github.com/) seperti berikut



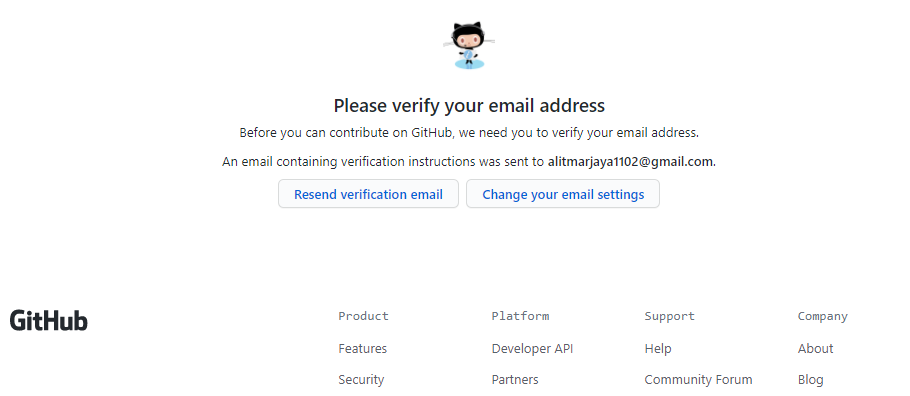
1. Lengkapi form sign up dengan data yang valid, lalu klik tombol Create an account



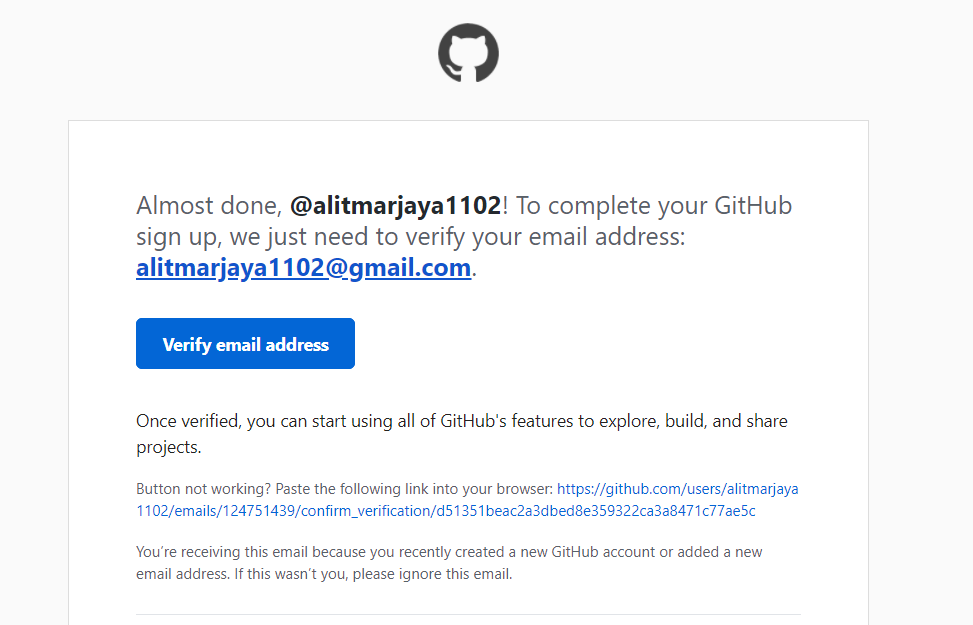
1. Lengkapi pilihan seperti gambar di bawah ini atau sesuai dengan basic kamu.



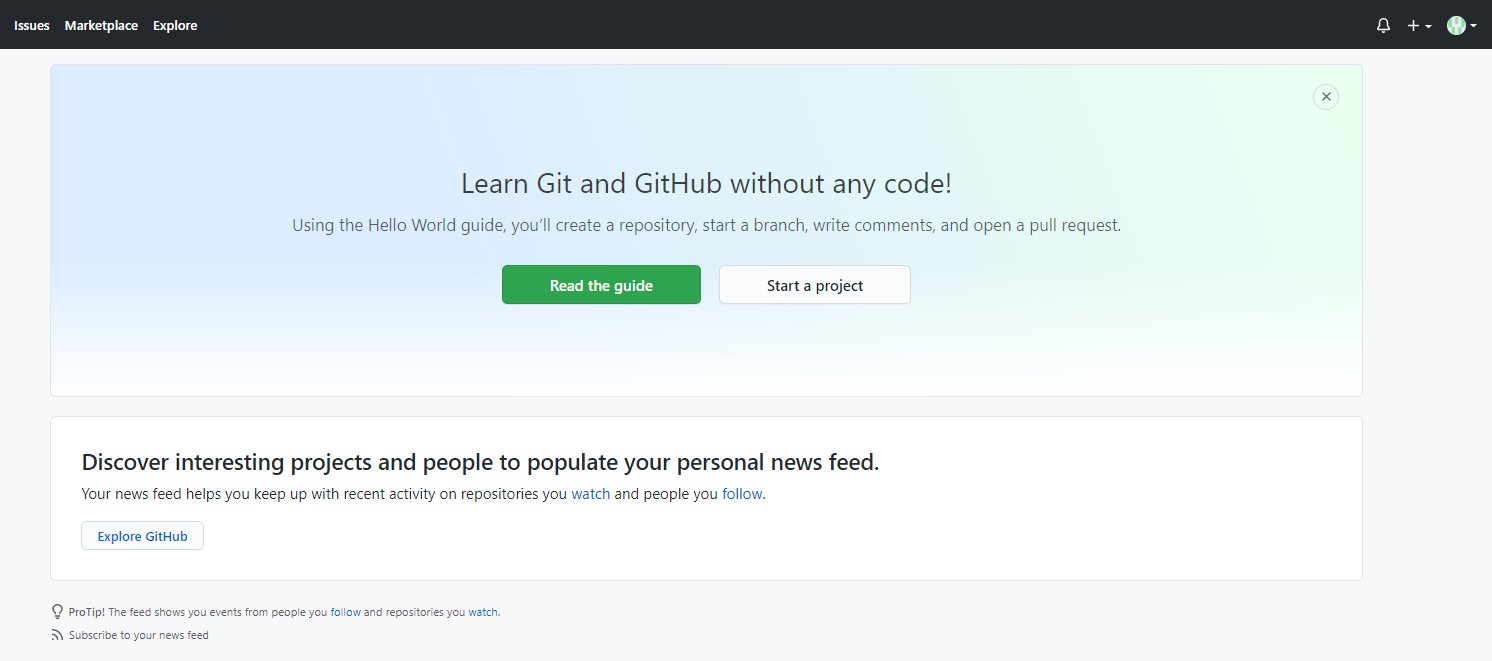
1. Silahkan buka email kamu dan klik link verifikasi untuk mengaktifkan akun.



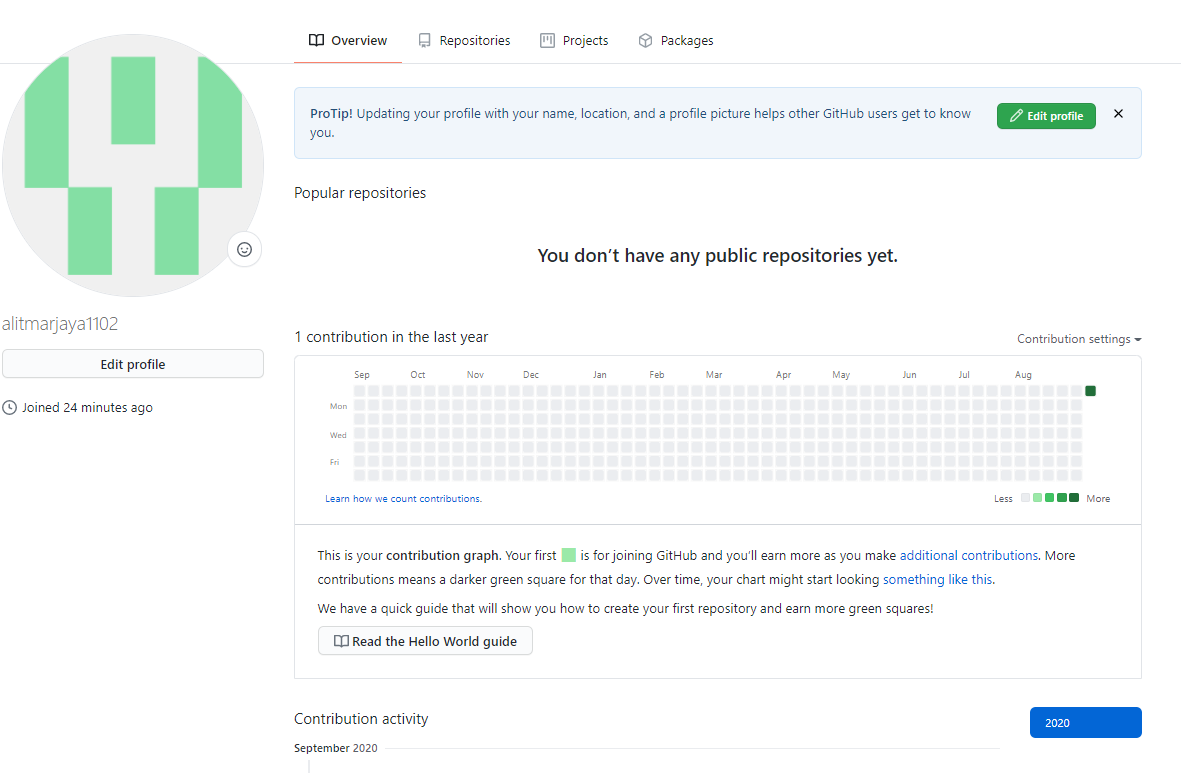
1. Klik verify email address.



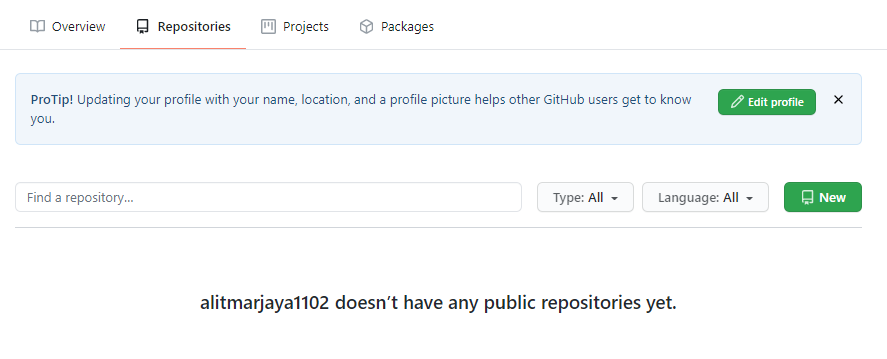
1. Selamat akun github telah berhasil dibuat.



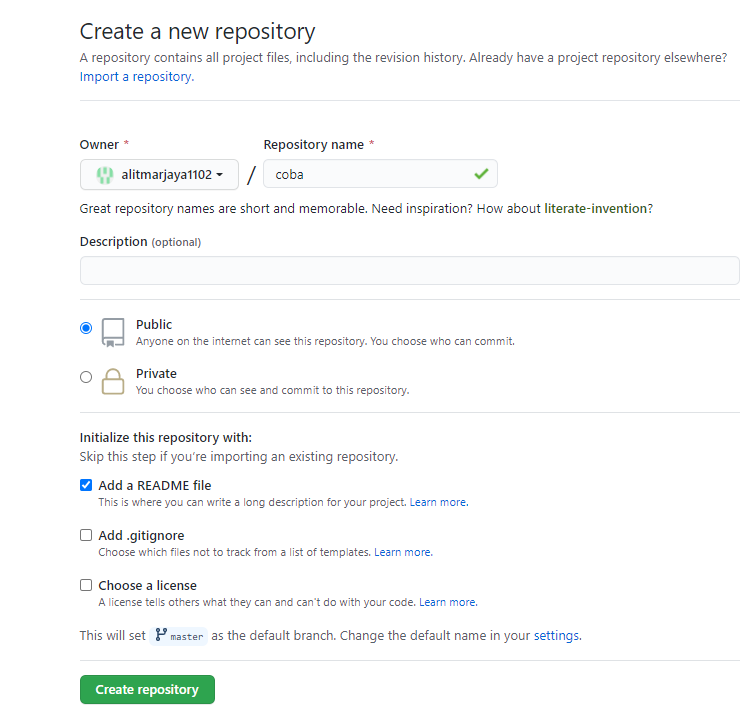
1. URL github kamu adalah <http://github.com/username>



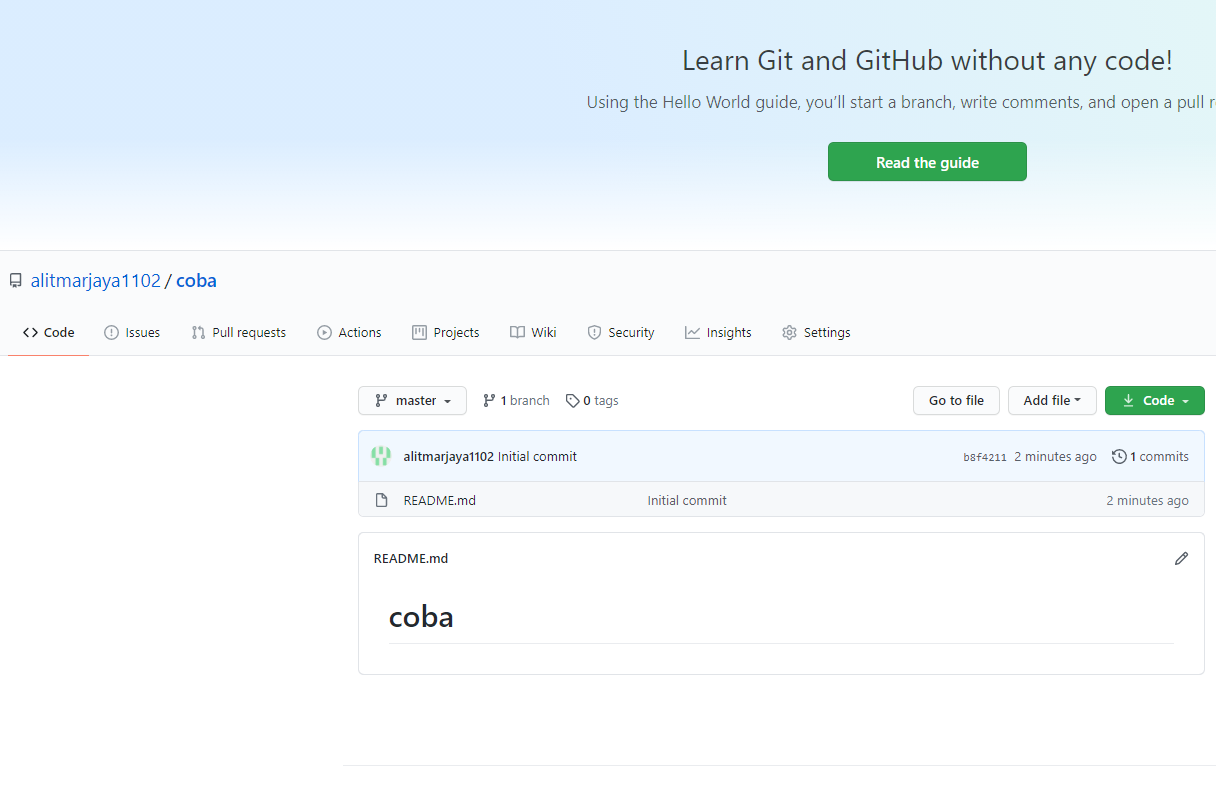
1. Selanjutnya buatlah repository seperti berikut dengan klik new



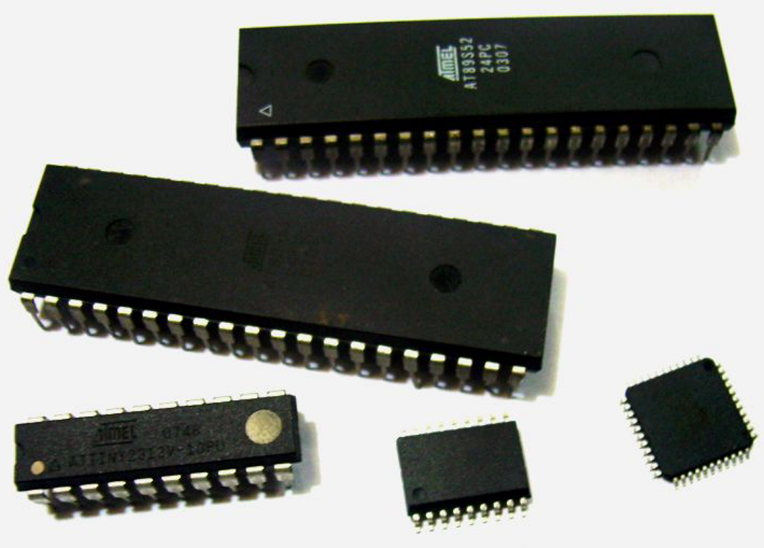
1. Selanjutnya isikan data sesuai dengan berikut :



1. Repository telah dibuat dengan nama **Coba**



1. Untuk mengupload file atau data yang ingin direpository dengan cara
   1. **Komponen-Komponen pada Alat** 
      1. **Pengenalan mengenai Microkontroler**



Mikrokontroler adalah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umunya dapat menyimpan program didalamnya. Mikrokontroler umumnya terdiri dari CPU (Central Processing Unit), memori, I/O tertentu dan unit pendukung seperti Analog-to-Digital Converter (ADC) yang sudah terintegrasi di dalamnya. Kelebihan utama dari mikrokontroler ialah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas. Mikrokontroler merupakan sebuah chip yang terdiri atas mikroprosesor, RAM, ROM, dan piranti I/O sehingga biasanya disebut single chip microcomputer. Salah satu keunggulan mikrokontroler adalah adanya RAM, ROM dan piranti I/O dalam sebuah chip sehingga tidak perlu menambahkan komponen-komponen tersebut secara fisik**.**

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output. Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Sekedar contoh, bayangkan diri Anda saat mulai belajar membaca dan menulis, ketika Anda sudah bisa melakukan hal itu Anda bisa membaca tulisan apapun baik buku, cerpen, artikel dan sebagainya, dan Andapun bisa pula menulis hal-hal sebaliknya. Begitu pula jika Anda sudah mahir membaca dan menulis data maka Anda dapat membuat program untuk membuat suatu sistem pengaturan otomatik menggunakan mikrokontroler sesuai keinginan Anda. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini.

Mikrokonktroler digunakan dalam produk dan alat yang dikendalikan secara automatis, seperti sistem kontrol mesin, remote controls, mesin kantor, peralatan rumah tangga, alat berat, dan mainan. Dengan mengurangi ukuran, biaya, dan konsumsi tenaga dibandingkan dengan mendesain menggunakan mikroprosesor memori, dan alat input output yang terpisah,

kehadiran mikrokontroler membuat kontrol elektrik untuk berbagai proses menjadi lebih ekonomis. Dengan penggunaan mikrokontroler ini maka :

* Sistem elektronik akan menjadi lebih ringkas
* Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi
* Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak

Namun demikian tidak sepenuhnya mikrokontroler bisa mereduksi komponen IC TTL dan CMOS yang seringkali masih diperlukan untuk aplikasi kecepatan tinggi atau sekedar menambah jumlah saluran masukan dan keluaran (I/O). Dengan kata lain, mikrokontroler adalah versi mini atau mikro dari sebuah komputer karena mikrokontroler sudah mengandung beberapa periferal yang langsung bisa dimanfaatkan, misalnya port paralel, port serial, komparator, konversi digital ke analog (DAC), konversi analog ke digital dan sebagainya hanya menggunakan sistem minimum yang tidak rumit atau kompleks.

Agar sebuah mikrokontroler dapat berfungsi, maka mikrokontroler tersebut memerlukan komponen eksternal yang kemudian disebut dengan sistem minimum. Untuk membuat sistem minimal paling tidak dibutuhkan sistem clock dan reset, walaupun pada beberapa mikrokontroler sudah menyediakan sistem clock internal, sehingga tanpa rangkaian eksternal pun mikrokontroler sudah beroperasi.

Untuk merancang sebuah sistem berbasis mikrokontroler, kita memerlukan perangkat keras dan perangkat lunak, yaitu:

1. Sistem minimal mikrokontroler
2. Software pemrograman dan kompiler, serta downloader

Yang dimaksud dengan sistem minimal adalah sebuah rangkaian mikrokontroler yang sudah dapat digunakan untuk menjalankan sebuah aplikasi. Sebuah IC mikrokontroler tidakakan berarti bila hanya berdiri sendiri. Pada dasarnya sebuah sistem minimal mikrokontroler AVR memiliki prinsip yang sama, yang terdiri dari 4 bagian, yaitu :

1. Prosesor, yaitu mikrokontroler itu sendiri
2. Rangkaian reset agar mikrokontroler dapat menjalankan program mulai dari awal
3. Rangkaian clock, yang digunakan untuk memberi detak pada CPU
4. Rangkaian catu daya, yang digunakan untuk memberi sumberdaya

Pada mikrokontroler jenis2 tertentu (AVR misalnya), poin2 pada no 2 ,3 sudah tersedia didalam mikrokontroler tersebut dengan frekuensi yang sudah diseting dari vendornya (biasanya 1MHz,2MHz,4MHz,8MHz), sehingga pengguna tidak perlu memerlukan rangkaian tambahan, namun bila ingin merancang sistem dengan spesifikasi tertentu (misal ingin komunikasi dengan PC atau handphone), maka pengguna harus menggunakan rangkaian clock yang sesuai dengan karakteristik PC atau HP tersebut, biasanya menggunakan kristal 11,0592 MHz, untuk menghasilkan komunikasi yang sesuai dengan baud rate PC atau HP tersebut.

**Sejarah Singkat Mikrokontroler**

Mikrokontroler adalah sebuah sistem komputer fungsional dalam sebuah chip. Di dalamnya terkandung sebuah inti prosesor, memori (sejumlah kecil RAM, memori program, atau keduanya), dan perlengkapan input output.

Dengan kata lain, mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi dan efektifitas biaya. Secara harfiahnya bisa disebut “pengendali kecil” dimana sebuah sistem elektronik yang sebelumnya banyak memerlukan komponen-komponen pendukung seperti IC TTL dan CMOS dapat direduksi/diperkecil dan akhirnya terpusat serta dikendalikan oleh mikrokontroler ini

Mikrokontroler pertama kali dikenalkan oleh Texas Instrument dengan seri TMS 1000 pada tahun 1974 yang merupakan mikrokontroler 4bit pertama. Mikrokontroler ini mulai dibuat sejak 1971. Merupakan mikrokomputer dalam sebuah chip, lengkap dengan RAM dan ROM. Kemudian, pada tahun 1976 Intel mengeluarkan mikrokontroler yang kelak menjadi populer dengan nama 8748 yang merupakan mikrokontroler 8bit, yang merupakan mikrokontroler dari keluarga MCS 48. Sekarang di pasaran banyak sekali ditemui mikrokontroler mulai dari 8bit sampai dengan 64 bit, sehingga perbedaan antara mikrokontroler dan mikroprosesor sangat tipis. Masing2 vendor mengeluarkan mikrokontroler dengan dilengkapi fasilitas2 yang cenderung memudahkan user untuk merancang sebuah sistem dengan komponen luar yang relatif lebih sedikit.

Saat ini mikrokontroler yang banyak beredar dipasaran adalah mikrokontroler 8bit varian keluarga MCS51(CISC) yang dikeluarkan oleh Atmel dengan seri AT89Sxx, dan mikrokontroler AVR yang merupakan mikrokontroler RISC dengan seri ATMEGA8535 (walaupun varian dari mikrokontroler AVR sangatlah banyak, dengan masing2 memiliki fitur yang berbeda2). Dengan mikrokontroler tersebut pengguna (pemula) sudah bisa membuat sebuah sistem untuk keperluan sehari-hari, seperti pengendali peralatan rumah tangga jarak jauh yang menggunakan remote control televisi, radio frekuensi, maupun menggunakan ponsel, membuat jam digital, termometer digital dan sebagainya.

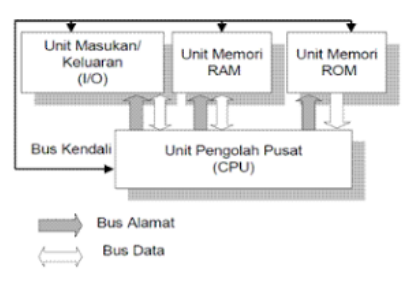
**Aplikasi Mikrokontroler**

Karena kemampuannya yang tinggi, bentuknya yang kecil, konsumsi dayanya yang rendah, dan harga yang murah maka mikrokontroler begitu banyak digunakan di dunia. Mikrokontroler digunakan mulai dari mainan anak-anak, perangkat elektronik rumah tangga, perangkat pendukung otomotif, peralatan industri, peralatan telekomunikasi, peralatan medis dan kedokteran, sampai dengan pengendali robot serta persenjataan militer. Terdapat beberapa keunggulan yang diharapkan dari alat-alat yang berbasis mikrokontroler (microcontroller-based solutions):

* + Kehandalan tinggi (high reliability) dan kemudahan integrasi dengan komponen lain (high degree of integration)
  + Ukuran yang semakin dapat diperkecil (reduced in size)
  + Penggunaan komponen dipersedikit (reduced component count) yang juga akan menyebabkan biaya produksi dapat semakin ditekan (lower manufacturing cost)
  + Waktu pembuatan lebih singkat (shorter development time) sehingga lebih cepat pula dijual ke pasar sesuai kebutuhan (shorter time to market) Konsumsi daya yang rendah (lower power consumption)

**Sistem Mikrokontroler**

Sebuah sistem minimum mikroprosesor memerlukan perangkat-perangkat seperti mikroprosesor, unit memori, unit masukan dan keluaran, dan rangkaian pendukung lain.



Dalam sistem mikroprosesor prinsip kerjanya adalah mengolah suatu data masukan,yang kemudian hasil olahan tersebut akan menghasilkan keluaran yang dikehendaki. Proses pengolahan datanya dapat difungsikan sesuai dengan instruksi yang diprogramkan. Masing-masing mikroprosesor memiliki bahasa pemrograman yang berbeda-beda. Namun secara prinsip, dasar dari tiap mikroprosesor adalah sama. Tiap mikroprosesor memiliki satu bus data, satu bus alamat dan satu bus kendali. Dalam mikroprosesor terdapat suatu unit untuk mengerjakan fungsi – fungsi logika dan aritmetika, register – register untuk menyimpan data sementara dan unit pengendalian.

Bus data terdiri biasanya 4, 8, 16 atau 32 jalur (bit), 64 bit, tergantung dari jenis mikroprosesornya. Bus data berfungsi memuat data dari dan ke mikroprosesor. Arah panah menunjukkan arah data dikirim/diterima. Bus alamat merupakan bus yang berisi alamat-alamat yang datanya akan dikirim/diterima oleh mikroprosesor. Bus kendali digunakan untuk mensinkronkan kerja antara mikroprosesor dengan dunia luar sistem. Pada beberapa aplikasi ada yang disebut dengan istilah jabat tangan, seperti misalnya pada penerapan hubungan dengan pencetak (printer). Dalam sistem kerjanya mikroprosesor didukung oleh unit memori (untuk menyimpan program tetap/sementara dan menyimpan data), unit masukan dan keluaran yang berfungsi sebagai antar muka dengan dunia luar. Catu daya, rangkaian pembangkit detak (clock), rangkaian pengawas sandi (address decoder), penyangga (buffer) dan penahan (latch) juga diperlukan mikroprosesor untuk mendukung operasi kerja sebagai satu rangkaian yang solid.

**Kelebihan Sistem Dengan Mikrokontroler**

Penggerak pada mikrokontoler menggunakan bahasa pemograman assembly dengan berpatokan pada kaidah digital dasar sehingga pengoperasian sistem menjadi sangat mudah dikerjakan sesuai dengan logika sistem (bahasa assembly ini mudah dimengerti karena menggunakan bahasa assembly aplikasi dimana parameter input dan output langsung bisa diakses tanpa menggunakan banyak perintah). Desain bahasa assembly ini tidak menggunakan begitu banyak syarat penulisan bahasa pemrograman seperti huruf besar dan huruf kecil untuk bahasa assembly tetap diwajarkan.

Mikrokontroler tersusun dalam satu chip dimana prosesor, memori, dan I/O terintegrasi menjadi satu kesatuan kontrol sistem sehingga mikrokontroler dapat dikatakan sebagai komputer mini yang dapat bekerja secara inovatif sesuai dengan kebutuhan sistem.

Sistem running bersifat berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk download perintah instruksi atau program. Langkah-langkah untuk download komputer dengan mikrokontroler sangat mudah digunakan karena tidak menggunakan banyak perintah.

Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem. Harga untuk memperoleh alat ini lebih murah dan mudah didapat.

**Fitur Kunci dari Mikrokontroller**

Agar dapat menentukan mikrokontroler mana yang dapat digunakan dengan paling maksimal untuk aplikasi yang akan dibuat, kita perlu mengetahui beberapa fitur kunci dari mikrokontroler dan apa saja fungsinya. Berikut adalah beberapa speksifikasi yang perlu diketahui dan harus perhitungkan ketika membaca dokumen data sheet dari suatu MCU:

* **Bits**: Mikrokontroler biasanya dijual berdasarkan jumlah bit yang mereka tawarkan. Ini berdampak pada kecepatan yang mampu mereka lakukan untuk komputasi yang tidak sepele.
* **RAM**: RAM adalah memori berkecepatan tinggi yang tidak menyimpan data ketika daya mati. Semua MCU memiliki sejumlah RAM, yang memungkinkannya untuk menyelesaikan berbagi aksi dengan cepat. Lebih banyak jumlahnya, lebih baik, namun jumlah yang besar tentunya meningkatkan biaya/harga dari MCU.
* **Flash:** Flash adalah memori komputer yang tetap menyimpan data meskipun daya mati. Tentunya fitur ini cukup penting, dan sangat berguna untuk penyimpanan offline.
* **GPIO**: GPIO singkatan dari pin untuk general-purpose input/output. Ini adalah pin-pin yang akan kita gunakan untuk menghubungkan sensor-sensor dan aktuator ke MCU dan Internet. Jumlah pin yang tersedia dapat beragam, mulai dari satu hingga ratusan, bergantung pada mikrokontrolernya.
* **Konektivitas**: Ini adalah bagaimana boardnya (dan juga aplikasinya) dapat terhubung ke Internet, baik melalui Wi-Fi, Ethernet, atau cara lainnya. Ini adalah aspek yang penting dari aplikasi sensor yang saling terhubung, sehingga topik ini akan diangkat dengan lebih detail pada bagian selanjutnya.
* **Konsumsi daya**: Konsumsi daya adalah hal yang sangat penting untuk penerapan sensor yang terhubung ke jaringan, khususnya ketika perangkatnya bergantung pada sumber daya dari baterai atau panel surya. Spesifikasi ini akan memberitahu kita seberapa rakuskah MCUnya secara bawaan dan apakah ia dapat mendukung teknik pemrograman yang sadar akan penggunaan daya.

Komunitas dan tools pengembangan: Tersedianya tools pemrograman dan pengembangan , dokumentasi, dan komunitas yang telah matang adalah salah satu aspek yang sangat penting untuk membantu kita membangun dan mengembangkan program yang akan dapat berjalan pada MCU yang kita pilih untuk aplikasi kita.

**Sistem Operasi Mikrokontroler**

Sekarang, mari bicara tentang sistem operasi yang berjalan di atas perangkat keras mikrokontroler. Seperti halnya personal komputer yang juga menjalankan sistem operasi seperti Windows, MCU juga menjalankan suatu sistem operasi.

Kita memiliki tiga pilihan utama:

* + 1. Bare Metal: Artinya bahwa, dengan kata lain, tanpa sistem operasi. Ini adalah pendekatan asli untuk bekerja dengan mikrokontroler, dan masih sangat populer karena sifatnya yang biayanya tepat sasaran dan efisien. Kekurangan utamanya adalah pilihan ini menyediakan dukungan yang minim bagi pengembang perangkat lunaknya.
    2. RTOS: Merukapan singkatan dari "Real-Time Operating System". Suatu sistem berbasis RTOS menyediakan garansi pasti dalam kaitannya dengan waktu dimana suatu operasi akan selesai. Ini sangat penting untuk mengkoordinasikan mesin-mesin fisik.
    3. Linux: Lebih mudah diprogram dan dihubungkan ke Internet. Ia berprilaku lebih mirip dengan "komputer beneran" sebagaimana rata-rata orang mungkin mengetahuinya, yang mana sangat cocok untuk berbagai alasan yang sebelumnya disampaikan. Namun, karena hal ini, ia tidak menyediakan garansi ketepatan pewaktuan apapun.

|  |  |  |
| --- | --- | --- |
| **Bare Metal** | **RTOS** | **Linux** |
| Pendekatan asli dan yang paling sederhana | Memberikan jaminan untuk waktu pemrosesan dengan peristiwa input / output | Sistem operasi open-source berbasis UNIX yang sangat populer, aslinya diperuntukan bagi personal komputer |
| Program berjalan langsung tanpa sistem operasi | Program berjalan di dalam sistem operasi | Jauh lebih aksesibel dan mudah untuk diprogram |
| Kode langsung bicara dengan komponen komputasi | Memiliki kemampuan untuk menjeda tugas dan memiliki kemampuan untuk memproritaskan eksekusi tugas yang berprioritas tinggi | Komunitas yang matang yang berisi orang-orang yang dapat membantu dan memberikan dukungan |
| Dukungan pemrograman yang terbatas | Cepat untuk disetup namun memerlukan banyak waktu untuk men-debug (melacak kesalahan) | Lebih susah untuk mendapatkan performa real-time |

**Pembagian Mikrokontroler**



Mikrokontroler dibagi menjadi 2 yaitu CISC (*Complex Instructions Set Computer*) dan RISC (*Reduce Instructions Set Computer*)

1. **CISC (*Complex Instructions Set Computer*)**

Complex instruction-set computing atau Complex Instruction-Set Computer (CISC; “Kumpulan instruksi komputasi kompleks”) adalah sebuah arsitektur dari set instruksi dimana setiap instruksi akan menjalankan beberapa operasi tingkat rendah, seperti pengambilan dari memory, operasi aritmetika, dan penyimpanan ke dalam memory, semuanya sekaligus hanya di dalam sebuah instruksi. Karakteristik CISC dapat dikatakan bertolak-belakang dengan RISC. Sebelum proses RISC didesain untuk pertama kalinya, banyak arsitek komputer mencoba menjembatani celah semantik”, yaitu bagaimana cara untuk membuat set-set instruksi untuk mempermudah pemrograman level tinggi dengan menyediakan instruksi “level tinggi” seperti pemanggilan procedure, proses pengulangan dan mode-mode pengalamatan kompleks sehingga struktur data dan akses array dapat dikombinasikan dengan sebuah instruksi. Karakteristik CISC yg “sarat informasi” ini memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Karena CISC inilah biaya pembuatan komputer pada saat itu (tahun 1960) menjadi jauh lebih hemat.

Memang setelah itu banyak desain yang memberikan hasil yang lebih baik dengan biaya yang lebih rendah, dan juga mengakibatkan pemrograman level tinggi menjadi lebih sederhana, tetapi pada kenyataannya tidaklah selalu demikian. Contohnya, arsitektur kompleks yang didesain dengan kurang baik (yang menggunakan kode-kode mikro untuk mengakses fungsi-fungsi hardware), akan berada pada situasi di mana akan lebih mudah untuk meningkatkan performansi dengan tidak menggunakan instruksi yang kompleks (seperti instruksi pemanggilan procedure), tetapi dengan menggunakan urutan instruksi yang sederhana.

Satu alasan mengenai hal ini adalah karena set-set instruksi level-tinggi, yang sering disandikan (untuk kode-kode yang kompleks), akan menjadi cukup sulit untuk diterjemahkan kembali dan dijalankan secara efektif dengan jumlah transistor yang terbatas. Oleh karena itu arsitektur -arsitektur ini memerlukan penanganan yang lebih terfokus pada desain prosesor. Pada saat itu di mana jumlah transistor cukup terbatas, mengakibatkan semakin sempitnya peluang ditemukannya cara-cara alternatif untuk optimisasi perkembangan prosesor. Oleh karena itulah, pemikiran untuk menggunakan desain RISC muncul pada pertengahan tahun 1970 (Pusat Penelitian Watson IBM 801 – IBMs)

*Contoh-contoh prosesor CISC adalah System/360, VAX, PDP-11, varian Motorola 68000 , dan CPU AMD dan Intel x86.*

**Sejarah CISC**

Sebelum proses RISC didesain untuk pertama kalinya, banyak arsitek komputer mencoba menjembatani celah semantik, yaitu bagaimana cara membuat set-set instruksi untuk mempermudah pemrograman level tinggi dengan menyediakan instruksi "level tinggi" seperti pemanggilan procedure, proses pengulangan dan mode-mode pengalamatan kompleks sehingga struktur data dan akses array dapat dikombinasikan dengan sebuah instruksi. Karakteristik CISC yg "sarat informasi" ini memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Karena CISC inilah biaya pembuatan komputer pada saat itu (tahun 1960) menjadi jauh lebih hemat.

Memang setelah itu banyak desain yang memberikan hasil yang lebih baik dengan biaya yang lebih rendah, dan juga mengakibatkan pemrograman level tinggi menjadi lebih sederhana, tetapi pada kenyataannya tidaklah selalu demikian. Contohnya, arsitektur kompleks yang didesain dengan kurang baik (yang menggunakan kode-kode mikro untuk mengakses fungsi-fungsi hardware), akan berada pada situasi di mana akan lebih mudah untuk meningkatkan performansi dengan tidak menggunakan instruksi yang kompleks (seperti instruksi pemanggilan procedure), tetapi dengan menggunakan urutan instruksi yang sederhana.

Satu alasan mengenai hal ini adalah karena set-set instruksi level-tinggi, yang sering disandikan (untuk kode-kode yang kompleks), akan menjadi cukup sulit untuk diterjemahkan kembali dan dijalankan secara efektif dengan jumlah transistor yang terbatas. Oleh karena itu arsitektur-arsitektur ini memerlukan penanganan yang lebih terfokus pada desain prosesor. Pada saat itu di mana jumlah transistor cukup terbatas, mengakibatkan semakin sempitnya peluang ditemukannya cara-cara alternatif untuk optimisasi perkembangan prosesor. Oleh karena itulah, pemikiran untuk menggunakan desain RISC muncul pada pertengahan tahun 1970 (Pusat Penelitian Watson IBM 801 - IBMs)

Istilah RISC dan CISC saat ini kurang dikenal, setelah melihat perkembangan lebih lanjut dari desain dan implementasi baik CISC dan CISC. Implementasi CISC paralel untuk pertama kalinya, seperti 486 dari Intel, AMD, Cyrix, dan IBM telah mendukung setiap instruksi yang digunakan oleh prosesor-prosesor sebelumnya, meskipun efisiensi tertingginya hanya saat digunakan pada subset x86 yang sederhana (mirip dengan set instruksi RISC, tetapi tanpa batasan penyimpanan/pengambilan data dari RISC). Prosesor-prosesor modern x86 juga telah menyandikan dan membagi lebih banyak lagi instruksi-instruksi kompleks menjadi beberapa "operasi-mikro" internal yang lebih kecil sehingga dapat instruksi-instruksi tersebut dapat dilakukan secara paralel, sehingga mencapai performansi tinggi pada subset instruksi yang lebih besar.

**Tujuan**

Tujuan utama dari arsitektur CISC adalah melaksanakan suatu instruksi cukup dengan beberapa baris bahasa mesin yang relatif pendek sehingga implikasinya hanya sedikit saja RAM yang digunakan untuk menyimpan instruksi-instruksi tersebut. Arsitektur CISC menekankan pada perangkat keras karena filosofi dari arsitektur CISC yaitu bagaimana memindahkan kerumitan perangkat lunak ke dalam perangkat keras.

**Karakteristik CISC**

* Sarat informasi memberikan keuntungan di mana ukuran program-program yang dihasilkan akan menjadi relatif lebih kecil, dan penggunaan memory akan semakin berkurang. Karena CISC inilah biaya pembuatan komputer pada saat itu (tahun 1960) menjadi jauh lebih hemat
* Dimaksudkan untuk meminimumkan jumlah perintah yang diperlukan untuk mengerjakan pekerjaan yang diberikan. (Jumlah perintah sedikit tetapi rumit) Konsep CISC menjadikan mesin mudah untuk diprogram dalam bahasa rakitan.

**Ciri-ciri**

* Jumlah instruksi banyak
* Banyak terdapat perintah bahasa mesin
* Instruksi lebih kompleks

**Kelebihan CISC:**

* Filosofi arsitektur CISC adalah memindahkan kerumitan software ke dalam hardware. Teknologi pembuatan IC saat ini memungkinkan untuk menamam ribuan bahkan jutaan transistor di dalam satu dice. Bermacam-macam instruksi yang mendekati bahasa pemrogram tingkat tinggi dapat dibuat dengan tujuan untuk memudahkan programmer membuat programnya.
* Beberapa prosesor CISC umumnya memiliki microcode berupa firmware internal di dalam chip-nya yang berguna untuk menterjemahkan instruksi makro. Mekanisme ini bisa memperlambat eksekusi instruksi, namun efektif untuk membuat instruksi-instruksi yang kompleks. Untuk aplikasi-aplikasi tertentu yang membutuhkan singlechip komputer, prosesor CISC bisa menjadi pilihan.
* CISC dimaksudkan untuk meminimumkan jumlah perintah yang diperlukan untuk mengerjakan pekerjaan yang diberikan. (Jumlah perintah sedikit tetapi rumit) Konsep CISC menjadikan mesin mudah untuk diprogram dalam bahasa rakitan.Dengan intruksi yang komplek prosesor CISC merupakan pendekatan dominan karena menghemat memori dibandingkan RISC.
* instruksi kompleks seperti CISC mempermudah dalam pembuatan program. Set instruksi yang lengkap diharapkan akan semakin membuat pengguna mikroprosesor leluasa menulis program dalam bahasa assembler yang mendekati bahasa pemrograman level tinggi.

**Kelemahan CISC :**

* Kompleksitas CPU : desain unit kontrol menjadi kompleks karena mempunyai set intruksiyang besar.
* Ukuran Sistem dan Biaya : mempunyai banyak sirkuit hardware menyebabkan CPU menjadikompleks. Hal ini meningkatkan biaya hardware pada sistem dan juga kebutuhan daya listrik.
* Kecepatan Clock: karena sirkuit yang besar maka propagation delay (tunda propagasi) lebihbesar dan karena waktu siklus CPU yang besar sehingga kecepatan clock efektif menurun.
* Keandalan : dengan hardware yang besar maka cenderung mudah terjadi kegagalan.
* Mantainability : Troubleshooting dan pendeteksian suatu kegagalan mengakibatkanpekerjaan menjadi besar karena besarnya sirkuit yang ada. Penemuan micro programming membantu menurunkan beban tersebut.

**Pengaplikasian CISC yaitu pada AMD dan Intel**

1. **RISC (*Reduce Instructions Set Computer*)**

RISC (Reduce Instruction Set Computer) atau komputasi set instruksi yang disederhanakan merupakan sebuah arsitektur komputer atau arsitektur komputasi modern dengan instruksi-instruksi dan jenis eksekusi yang paling sederhana. Arsitektur ini digunakan pada komputer dengan kinerja tinggi, seperti komputer vector. Desain ini juga diimplementasikan pada prosesor komputer lain, seperti pada beberapa mikroprosesor intel 960, Itanium(IA64) dari Intel Coorporation. Selain itu RISC juga umum dipakai pada Advanced RISC Machine(ARM) dan StrongARM.

RISC, yang jika diterjemahkan berarti "Komputasi Kumpulan Instruksi yang Disederhanakan", merupakan sebuah arsitektur komputer atau arsitektur komputasi modern dengan instruksi-instruksi dan jenis eksekusi yang paling sederhana. Arsitektur ini digunakan pada komputer dengan kinerja tinggi, seperti komputer vektor. Selain digunakan dalam komputer vektor, desain ini juga diimplementasikan pada prosesor komputer lain, seperti pada beberapa mikroprosesor Intel 960, Itanium (IA64) dari Intel Corporation, Alpha AXP dari DEC, R4x00 dari MIPS Corporation,PowerPC dan Arsitektur POWER dari International Business Machine. Selain itu, RISC juga umum dipakai pada Advanced RISC Machine (ARM) dan StrongARM (termasuk di antaranya adalah Intel XScale), SPARC dan UltraSPARC dari Sun Microsystems, serta PA-RISC dari Hewlett-Packard.

**Sejarah RISC**

Reduced Instruction Set Computing (RISC) atau "Komputasi set instruksi yang disederhanakan" pertama kali digagas oleh John Cocke, peneliti dari IBM di Yorktown, New York pada tahun 1974 saat ia membuktikan bahwa sekitar 20% instruksi pada sebuah prosesor ternyata menangani sekitar 80% dari keseluruhan kerjanya. Komputer pertama yang menggunakan konsep RISC ini adalah IBM PC/XT pada era 1980-an. Istilah RISC sendiri pertama kali dipopulerkan oleh David Patterson, pengajar pada University of California di Berkeley.

**Perkembangan RISC**

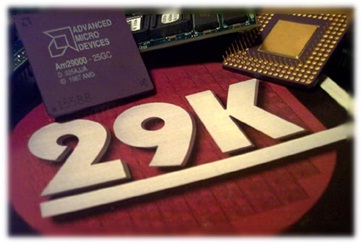
Ide dasar prosesor RISC sebenarnya bisa dilacak dari apa yang disarankan oleh Von Neumann pada tahun 1946. Von Neumann menyarankan agar rangkaian elektronik untuk konsep logika diimplementasikan hanya bila memang diperlukan untuk melengkapi sistem agar berfungsi atau karena frekuensi penggunaannya cukup tinggi (Heudin, 1992 : 18). Jadi ide tentang RISC, yang pada dasarnya adalah untuk menyederhanakan realisasi perangkat keras prosesor dengan melimpahkan sebagian besar tugas kepada perangkat lunaknya, telah ada pada komputer elektronik pertama. Seperti halnya prosesor RISC, komputer elektronik pertama merupakan komputer eksekusi-langsung yang memiliki instruksi sederhana dan mudah didekode.

Hal yang sama dipercayai juga oleh Seymour Cray, spesialis pembuat superkomputer. Pada tahun 1975, berdasarkan kajian yang dilakukannya, Seymour Cray menyimpulkan bahwa penggunaan register sebagai tempat manipulasi data menyebabkan rancangan instruksi menjadi sangat sederhana. Ketika itu perancang prosesor lain lebih banyak membuat instruksi-instruksi yang merujuk ke memori daripada ke register seperti rancangan Seymour Cray. Sampai akhir tahun 1980-an komputer-komputer rancangan Seymour Cray, dalam bentuk superkomputer seri Cray, merupakan komputer-komputer dengan kinerja sangat tinggi.

Pada tahun 1975, kelompok peneliti di IBM di bawah pimpinan George Radin, memulai merancang komputer berdasar konsep John Cocke. Berdasarkan saran John Cocke, setelah meneliti frekuensi pemanfaatan instruksi hasil kompilasi suatu program, untuk memperoleh prosesor berkinerja tinggi tidak perlu diimplementasikan instruksi kompleks ke dalam prosesor bila instruksi tersebut dapat dibuat dari instruksi-instruksi sederhana yang telah dimilikinya. Kelompok IBM ini menghasilkan komputer 801 yang menggunakan instruksi format-tetap dan dapat dieksekusi dalam satu siklus detak. Komputer 801 yang dibuat dengan teknologi ECL (emitter-coupled logic) , 32 buah register, chace terpisah untuk memori dan instruksi ini diselesaikan pada tahun 1979. Karena sifatnya yang eksperimental, komputer ini tidak dijual di pasaran.

**Prosesor RISC Berkeley**

Kelompok David Patterson dari Universitas California memulai proyek RISC pada tahun 1980 dengan tujuan menghindari kecenderungan perancangan prosesor yang perangkat instruksinya semakin kompleks sehingga memerlukan perancangan rangkaian kontrol yang semakin rumit dari waktu ke waktu. Hipotesis yang diajukan adalah bahwa implementasi instruksi yang kompleks ke dalam perangkat instruksi prosesor justru berdampak negatif pemakaian instruksi tersebut dalam kebanyakan program hasil komplikasi (Heudin, 1992 : 22). Apalagi, instruksi kompleks itu pada dasarnya dapat disusun dari instruksi-instruksi sederhana yang telah dimiliki.



*Turunan Processor Risc Berkeley*

Rancangan prosesor RISC-1 ditujukan untuk mendukung bahasa C, yang dipilih karena popularitasnya dan banyaknya pengguna. Realisasi rancangan diselesaikan oleh kelompok Patterson dalam waktu 6 bulan. Fabrikasi dilakukan oleh MOVIS dan XEROX dengan menggunakan teknologi silikon NMOS (N-channel Metal-oxide Semiconductor) 2 mikron. Hasilnya adalah sebuah cip rangkaian terpadu dengan 44.500 buah transistor. Cip RISC-1 selesai dibuat pada musim panas dengan kecepatan eksekusi 2 mikrosekon per instruksi (pada frekuensi detak 1,5 MHz), 4 kali lebih lambat dari kecepatan yang ditargetkan. Tidak tercapainya target itu disebabkan terjadinya sedikit kesalahan perancangan, meskipun kemudian dapat diatasi dengan memodifikasi rancangan assemblernya.

Berdasarkan hasil evaluasi, meskipun hanya bekerja pada frekuensi detak 1,5 MHz dan mengandung kesalahan perancangan, RISC-1 terbukti mampu mengeksekusi program bahasa C lebih cepat dari beberapa prosesor CISC, yakni MC68000, Z8002, VAX-11/780, dan PDP-11/70.

Hampir bersamaan dengan proses fabrikasi RISC-1, tim Berkeley lain mulai bekerja untuk merancang RISC-2. Cip yang dihasilkan tidak lagi mengandung kesalahan sehingga mencapai kecepatan operasi yang ditargetkan, 330 nanosekon tiap instruksi (Heudin, 1992 : 27-28). RISC-2 hanya memerlukan luas cip 25% dari yang dibutuhkan RISC-1 dengan 75% lebih banyak register. Meskipun perangkat instruksi yang ditanamkan sama dengan perangkat instruksi yang dimiliki RISC-1, tetapi di antara keduanya terdapat perbedaan mikroarsitektur perangkat kerasnya. RISC-2 memiliki 138 buah register yang disusun sebagai 8 jendela register, dibandingkan dengan 78 buah register yang disusun sebagai 6 jendela register. Selain itu, juga terdapat perbedaan dalam hal organisasi alur-pipa (pipeline) . RISC-1 memiliki alur-pipa dua tingkat sederhana dengan penjeputan (fetch) dan eksekusi instruksi yang dibuat tumpang-tindih, sedangkan RISC-2 memiliki 3 buah alur-pipa yang masing-masing untuk penjemputan instruksi, pembacaan operan dan eksekusinya, dan penulisan kembali hasilnya ke dalam register.

Sukses kedua proyek memacu tim Berkeley untuk mengerjakan proyek SOAR (Smalltalk on RISC) yang dimulai pada tahun 1983. Tujuan proyek ini adalah untuk menjawab pertanyaan apakah arsitektur RISC bekerja baik dengan bahasa pemrograman Smalltalk? Jadi proyek SOAR ini merupakan upaya pertama menggunakan pendekatan RISC untuk pemrosesan simbolik. Versi pertama mikroprosesor SOAR diimplementasikan dengan menggunakan teknologi NMOS 4 mikron. Cip yang dihasilkan memiliki 35.700 buah transistor dan bekerja dengan kecepatan 300 nanosekon tiap instruksi. Versi kedua yang dirancang pada 1984-1985 menggunakan teknologi CMOS (Complementary Metal-oxide Semiconductor). Beberapa prosesor berarsitektur RISC banyak yang dipengaruhi oleh rancangan mikroprosesor SOAR, misalnya mikroprosesor SPARC (dari Sun Microsystems Inc.) dan KIM20 yang dirancang Departemen Pertahanan Perancis.



*Mikroprosesor SPARC*

Mengikuti proyek SOAR, kelompok Berkeley kemudian mengerjakan proyek SPUR (Symbolic Processing Using RISC) yang dimulai tahun 1985. Proyek SPUR bertujuan untuk merancang stasiun-kerja(workstation) multiprosesor sebagai bagian dari riset tentang pemrosesan paralel (Robinson, 1987 : 145). Selain itu, proyek SPUR juga melakukan penelitian tentang rangkaian terpadu, arsitektur komputer, sistem operasi, dan bahasa pemrograman. Sistem prosesor SPUR dibangun dengan 6-12 prosesor berkinerja tinggi yang dihubungkan satu sama lain, serta dihubungkan dengan memori dan peranti masukan/keluaran melalui Nubus yang telah dimodifikasi. Unjuk kerja sistem diperbaiki dengan menambahkan chace sebesar 128 kilobyte pada tiap prosesor untuk mengurangi kepadatan lalu lintasdata pada bus dan mengefektifkan pengaksesan.

**Prosesor RISC Stanford**

Sementara proyek RISC-1 dan RISC-2 dilakukan kelompok Patterson di Universitas California, pada tahun 1981 itu juga John Hennessy dari Universitas Stanford mengerjakan proyek MIPS (Microprocessor without Interlocked Pipeline Stages) . Pengalaman riset tentang optimasi kompilator digabungkan dengan teknologi perangkat keras RISC merupakan kunci utama proyek MIPS ini. Tujuan utamanya adalah menghasilkan cip mikroprosesor serbaguna 32-bit yang dirancang untuk mengeksekusi secara efisien kode-kode hasil kompilasi (Heudin, 1992: 34).

Perangkat instruksi prosesor MIPS terdiri atas 31 buah instruksi yang dibagi menjadi 4 kelompok, yakni kelompok instruksi isi dan simpan, kelompok instruksi operasi aritmetika dan logika, kelompok instruksi pengontrol, dan kelompok instruksi lain-lain. MIPS menggunakan lima tingkat alur-pipa tanpa perangkat keras saling-kunci antar alur-pipa tersebut, sehingga kode yang dieksekusi harus benar-benar bebas dari konflik antar alur-pipa.

Direalisasi dengan teknologi NMOS 2 mikron, prosesor MIPS yang memiliki 24.000 transistor ini memiliki kemampuan mengeksekusi satu instruksi setiap 500 nanodetik. Karena menggunakan lima tingkat alur-pipa bagian kontrol prosesor MIPS ini menyita luas cip dua kali lipat dibanding dengan bagian kontrol pada prosesor RISC. MIPS memiliki 16 register dibandingkan dengan 138 buah register pada RISC-2. Hal ini bukan masalah penting karena MIPS memang dirancang untuk mebebankan kerumitan perangkat keras ke dalam perangkat lunak sehingga menghasilkan perangkat keras yang jauh lebih sederhana dan lebih efisien. Perangkat keras yang sederhana akan mempersingkat waktu perancangan, implementasi, dan perbaikan bila terjadi kesalahan.

Sukses perancangan MIPS dilanjutkan oleh tim Stanford dengan merancang mikroprosesor yang lebih canggih, yakni MIPS-X. Perancangan dilakukan oleh tim riset MIPS sebelumnya ditambah 6 orang mahasiswa, dan dimulai pada musim panas tahun 1984. Rancangan MIPS-X banyak diperbaruhi oleh MIPS dan RISC-2 dengan beberapa perbedaan utama :

* Semua instruksi MIPS-X merupakan operasi tunggal dan dieksekusi dalam satu siklus detak
* Semua instruksi MIPS-X memiliki format tetap dengan panjang instruksi 32-bit
* MIPS-X dilengkapi pendukung koprosesor yang efisien dan sederhana
* MIPS-X dilengkapi pendukung untuk digunakan sebagai prosesor dasar dalam sistem multiprosesor memori-bersama (shared memory)
* MIPS-X dilengkapi chace instruksi dalam-cip yang cukup besar (2 kilobyte) MIPS-X difabrikasi dengan teknologi CMOS 2 mikron.

Sama seperti MIPS, MIPS-X merupakan prosesor dengan alur-pipa tanpa saling-kunci (interlock) perangkat keras. Perangkat lunaknya dirancang untuk mengikuti pewaktuan instruksi agar tidak terjadi konflik antar alur-pipa (Heudin, 1992 : 36-37).

Cip pertama yang dihasilkan bekerja baik dengan detak 16 MHz, lebih rendah dari target yang dicanangkan setinggi 20 MHz, akibat tidak sempurnanya instruksi percabangan. Versi 25 MHz dibuat dengan menggunakan teknologi CMOS 1,6 mikron. Ditambah dengan chace yang diintregrasikan pada cip prosesor, MIPS-X berisi hampir 150.000 transistor di atas keping seluas 8 x 8,5 mm.

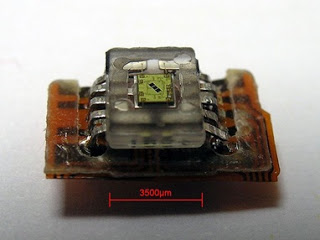


*MIPS X Motherboard Block*

**Arah Perkembangan Prosesor RISC**

Kebanyakan riset tentang prosesor RISC ditujukan untuk memperbaiki kinerja sistem komputer secara keseluruhan. Analisis yang mendalam menunjukkan bahwa ada dua arah perlembangan penting prosesor RISC yaitu upaya ke arah pemanfaatan teknologi proses yang mampu menghasilkan prosesor cepat, misalnya teknologi bipolar ECL (emitter-coupled logic) serta pemanfaatan bahan semikonduktor GaAs(galium arsenida). Arah lain adalah upaya untuk merancang arsitektur multiprosesor dan mengintegrasikan unit-unit fungsional pendukung pemrosesan paralel dalam satu cip.

**Cip-cip RISC galium Arsenida**



*Chip Galium Arsenida*

Galium Arsenida dapat digunakan untuk menggantikan silikon dalam beberapa rangkaian terpadu untuk pemakaian khusus. Keunggulan bahan GaAs dibandingkan silikon adalah ketahanannya terhadap radiasi, dan ketahanannya terhadap panas, serta kecepatan mobilitas elektronnya. Karena elektron dapat bergerak lebih cepat dalam bahan GaAs, maka cip yang dibuat dengan bahan ini berpotensi untuk bekerja lebih cepat (Jonhsen, 1984 : 46; Robinson, 1990 : 251-254). Salah satu kendala pengembangan cip berbahan GaAs adalah sulitnya penanganan bahan ini dibanding dengan bahan silikon karena perancang belum banyak pengalaman dengan bahan GaAs. Meskipun demikian, teknologi yang dikuasai saat ini telah memungkinkan untuk membuat rangkaian terintegrasi dengan tingkat kerapatan cukup tinggi untuk merancang prosesor RISC.

Didorong oleh kebutuhan untuk merancang prosesor berkecepatan tinggi dan tahan terhadap radiasi sesuai dengan spesifikasi yang dibutuhkan Departemen Pertahanan Amerika Serikat, maka DARPA (Defense Advanced Research Projects Agency) memberikan dana kepada Texas Instruments (TI), RCA, dan McDonnell-Douglas, untuk mengembangkan dan merancang prosesor RISC dari bahan GaAs. Agar memiliki kinerja yang tinggi, DARPA menghendaki unit prosesor sentral (central processing unit, CPU) dirancang dalam cip tunggal, seperti prosesor MIPS yang pengembangannya juga dibiayai DARPA. Ditargetkan prosesor tersebut akan dapat dijalankan dengan detak berfrekuensi 200 MHz. Ini berarti target kecepatan kerjanya adalah 200 MIPS (million instructions per second, juta instruksi per detik), karena pada prosesor RISC satu instruksi dieksekusi dalam satu siklus detak.

Sistem yang dipilih terdiri dari seperangkat cip, yakni, CPU, FCOP (floating point coprocessor) , MMU(memory management unit) dan chace. Agar bisa merealisasi CPU dalam satu cip, TI berupaya mengurangi rangkaian pengontrol sebanyak mungkin untuk memberi lebih banyak tempat bagi register-register. Perangkat instruksi dikembangkan berdasarkan simulasi statistik dan evaluasi atas prosesor RISC Berkeley maupun MIPS Stanford. Seperti halnya MIPS, sekali program telah dikomplikasi ke dalam perangkat instruksi inti (yakni level tengah antara perangkat-intruksi bergantung perangkat-keras dengan bahasa pemrograman tingkat tinggi), suatu penerjemah bergantung perangkat-keras akan mengubah kode ke dalam perangkat instruksi bahasa mesin dan melakukan langkah-langkah optimasi. Perangkat instruksi yang dimiliki prosesor ini dibagi menjadi tiga bagian yakni 29 buah instruksi CPU, 31 buah instruksi FCOP, serta 6 buah instruksi MMU.

Prosesor yang dihasilkan memiliki unjuk kerja nominal 200 MIPS, tetapi angka faktualnya harus dikurangi dengan 32% akibat penyisipan instruksi NOP (no operation) dan dikurangi 32% lagi karena keterbatasan lebar ban memori. Angka faktual kinerja prosesor RISC GaAs ini kira-kira 91 MIPS (million instruction per second).

Pada waktu yang sama dengan pengembangan mikroprosesor RISC GaAs, McDonnell-Douglas juga mulai mengembangkan mikroprosesor RISC berdasarkan teknologi JFET tipe-penyambungan (enhancement-type junction field-effect transistor) DCFL (direct coupled FET logic) dengan bahan GaAs. Cip yang diberi nama MD484 sangat dipengaruhi oleh hasil rancangan MIPS dari Universitas Stanford.

Karena saat itu teknologi GaAs hanya mampu mengintegrasikan transistor dalam jumlah yang terbatas, maka hanya ditargetkan sejumlah 25.000 buah transistor dalam satu cip. Di dalam mikroprosesor ditanamkan 32 buah register masing-masing 32-bit dengan perangkat instruksi sangat mirip dengan yang dimiliki MIPS.

Salah satu keputusan sulit dalam perancangan adalah masalah memilih jumlah dan tipe alur-pipa eksekusi. Penambahan jumlah alur-pipa menjadi lima atau enam dengan penambahan tingkat alur-pipa untuk akses memori, akan memberi lebih banyak waktu pengaksesan memori sehingga memudahkan perancangan sistem memori. Akan tetapi, alur-pipa yang panjang akan menambah tundaan pencabangan sehingga memperlambat waktu eksekusi. Kerugian kinerja akibat penyisipan instruksi NOP adalah 20-30% untuk alur-pipa enam tingkat dan kira-kira setengahnya untuk alur-pipa lima tingkat relatif terhadap alur-pipa empat tingkat. Akhirnya, kelompok McDonnell-Douglas memutuskan untuk menggunakan empat tingkat alur-pipa. Untuk mengeksekusi operasi aritmetika floating point, McDonnell Douglas juga merancang cip koprosesor floating point. Cip CPU yang selesai dibuat dan diuji pada tahun 1987, mampu mengeksekusi instruksi dalam 16,5 nanosekon dan memberikan kecepatan operasi 60 MIPS (million instructions per second).

Proyek perancangan prosesor RISC GaAs lain dilakukan oleh RCA pada tahun 1989. Prosesor 32-bit rancangan RCA ini direncanakan diimplementasikan dengan GaAs VLSI (very large scale integration) . RCA mengatasi masalah yang dihadapi dalam perancangan cip GaAs ini dengan cara yang berbeda dari yang dilakukan McDonnell Douglas maupun Texas Instruments. Berbeda dengan kebanyakan prosesor RISC, format instruksinya tidak tunggal melainkan menggunakan format satu dan dua kata. Rancangan RCA ini menggunakan 9 tingkat alur-pipa dengan dua periode tak-aktif masing-masing 2 siklus tunggu, pertama berkaitan dengan penjemputan instruksi dan kedua berkaitan dengan penjemputan operan untuk operasiload.

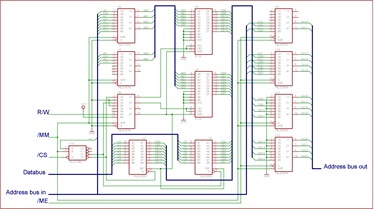
Kelompok riset di Universitas Michigan juga dilaporkan berhasil membuat prosesor RISC dari bahan galium arsenida berkecepatan tinggi di atas cip berukuran 32-bit yang dihasilkan diimplementasikan di atas cip berukuran 13,9 x 7,8 mm dengan 160.000 transistor. Di dalam cip diintegrasikan bagian ALU (arithmetic and logic unit) , 32 buah register, dan 32 byte chace instruksi. Karena kecilnya chace yang dimiliki, pemakai prosesor ini dapat menambahkan chace eksternal melalui kecepatan tinggi misalnya dengan SRAM (static random access memory) berteknologi ECL. Cip ini bekerja baik dengan frekuensi detak 200 MHz.

Ada beberapa permasalahan dalam perancangan komputer cepat dengan GaAs. Pertama, adalah terbatasnya tingkat integrasi fungsi logika yang bisa diimplementasikan. Kedua, adalah tingginya perbandingan antara waktu pengaksesan memori di luar cip dengan akses data di dalam cip. SODIMA S.A. mengusulkan arsitektur 4-tingkat 32-bit untuk diintegrasikan dengan menggunakan teknologi sel standar. Tim SODIMA juga merancang arsitektur chacechace kecil berkecepatan tinggi (4-kilobyte dengan waktu akses 3 nanosekon) dikombinasikan dengan chacebesar tetapi lebih lambat (128 kilobyte dengan waktu akses 25 nanosekon) untuk mendapatkan kinerja 100 MIPS. dua tingkat berdasarkan pada

**Cip RISC lain**

Advanced Micro Devices (AMD) memperkenalkan produk RISC-nya pada tahun 1987, yang diberi nama Am29000. Dengan eksekusi siklus tunggal, prosesor yang memiliki detak berfrekuensi 25MHz ini memiliki kecepatan proses 17 MIPS untuk program bahasa C. Ada dua tingkat optimasi kinerja yang dilakukan dalam perancangan Am29000. Pertama, prosesor ini memiliki jumlah register cukup banyak (192 buah) yang dapat difungsikan sebagai chace(stack) instruksi saat suatu prosedur dipanggil atau sebagai kelompok register, masing-masing terdiri atas 16 buah register. Rancangan khusus dalam Am29000 adalah chaceuntuk target pencabangan yang mampu menyimpan 128 instruksi. Cara ini memungkinkan alur-pipa tetap terisi tanpa adanya penundaan sebagai akibat dari operasi percabangan yang berturutan (Heudin, 1992 : 104). untuk menetapkan tumpukan

Selain AMD, Intel yang dikenal sebagai pemasok mikroprosesor CISC keluarga-86, juga memproduksi cip mikroprosesor RISC yang diberi nama 80860 pada tahun 1989. Dengan mengintegrasikan lebih dari sejuta transistor, 80860 berisi teras RISC (RISC core) , koprosesor atau unit floating point, MMU (memory management unit) , unit grafik, dan chace terpisah untuk data dan instruksi. Keberadaan MMU dan teras RISC memungkinkan 80860 menjalankan sistem operasi multitasking. Koprosesornya mendukung aplikasi pemodelan, pengolahan suara, simulasi, dan perancangan berbantuan komputer (Margulis, 1989 : 333). Teras RISC memiliki empat tingkat alur-pipa yang meliputi tingkat penjemputan, dekode, eksekusi, dan penulisan instruksi.



*MMU dengan 8 bit CPU*

Keistimewaannya, prosesor ini dirancang agar pemrogram dapat memilih sendiri mode eksekusi yang diperlukan, yakni instruksi-tunggal dan instruksi-ganda. Instruksi tunggal merupakan mode eksekusi tradisional, dengan penjemputan instruksi berturutan. Pemberian alur-pipa memungkinkan instruksi berturutan tersebut saling tumpang-tindih sehingga beberapa instruksi berada di beberapa tingkat alur-pipa untuk dieksekusi kapan saja. Dengan mode instruksi-ganda, mikroprosesor 80860 menerapkan lebih dari sekedar strategi alur-pipa. Mode ini memungkinkan dijalankannya dua instruksi sekaligus, satu untuk teras RISC dan satu untuk koprosesor. Koprosesor atau unit floating point menampilkan hasil operasi setiap satu siklus detak dan memungkinkan diselesaikannya dua operasi sekaligus, misalnya operasi penjumlahan dan perkalian. Dengan mengkombinasikan mode instruksi-ganda dan mode operasi-ganda, pemrogram dapat melakukan tiga operasi sekaligus setiap satu siklus detak.

Cip RISC dengan detak berfrekuensi lebih dari 300 MHz dilaporkan telah dibuat oleh Digital Equipment Corp. (DEC). Cip yang dirancang dengan teknologi bipolar ECL itu mengimplementasikan 468.000 buah transistor dan 206.000 resistor di atas keping berukuran 15,4 x 12,6 mm. Pada kondisi terburuk, yakni dengan tegangan catu daya -5,2 volt, prosesor ini mampu dijalankan dengan detak internal berfrekuensi 275 MHz sedangkan dalam kondisi puncaknya (dengan tegangan catu daya -3,9 volt) dapat beroperasi pada frekuensi detak 335 MHz. Pembangkit detak eksternal memiliki frekuensi 80 MHz yang kemudian dilipatkan oleh rangkaian PLL (phase-locked loop) menjadi 1X - 8X. Masalah besar yang timbul dengan teknologi bipolar ECL ini adalah kebutuhan daya yang cukup besar, yakni mencapai 115 watt. Hal ini menyebabkan timbulnya panas berlebihan dalam cip. Untuk mengatasinya, DEC menambahkan termosifon (penghambur panas berbentuk silinder bersirip dari tembaga) di atas kemasan cip agar suhu dalam cip terjaga tidak lebih dari 100o C.

**Prospek Arsitektur RISC di Masa Mendatang**

Perkembangan menarik terjadi pada tahun 1993 ketika aliansi tiga perusahaan terkemuka, IBM, Apple, dan Motorola memperkenalkan produk baru mereka yakni PowerPC 601, suatu mikroprosesor RISC 64-bit yang dirancang untuk stasiun kerja (workstation) atau komputer personal. Menarik, karena kemunculan PowerPC 601 dimaksudkan untuk memberikan alternatif bagi dominasi prosesor CISC keluarga-86 Intel dalam komputer rumahan. Popularitas prosesor keluarga-86 didukung oleh harganya yang murah dan banyaknya program aplikasi yang dapat dijalankan dengan prosesor ini. Untuk itu, prosesor PowerPC dijual dengan harga yang cukup bersaing dibandingkan dengan pentium, yakni prosesor buatan Intel mutakhir saat itu (Thompson, 1993 : 64). Perkembangan teknologi emulasi yang memungkinkan prosesor RISC menjalankan sistem operasi yang sama dengan prosesor CISC keluarga-86 diperkirakan akan membuat prosesor RISC, terutama PowerPC 601, banyak digunakan di dalam komputer-komputer personal.

PowerPC 601 memiliki 32 buah register serbaguna 32-bit dan 32 buah 64-bit register floating-point. Untuk menyimpan sementara data dan instruksi sebelum dieksekusi, PowerPC 601 memiliki 32-kilobyte chaceuntuk data dan instruksi bersama-sama. Teras PowerPC 601 terdiri dari tiga unit eksekusi dengan alur-pipa yang independen, yakni unit pemroses bilangan bulat (IU, integer unit), unit floating-point (FPU, floating processing unit), dan unit pemroses operasi percabangan (BPU, branch processing unit) yang mampu mengeksekusi tiga instruksi sekaligus .



*IBM Power PC 601*

Perkembangan menarik juga nampak dengan diadopsinya sebagian arsitektur RISC ke dalam prosesor CISC yang dikenal dengan sebutan arsitektur hibrid CISC/RISC. Intel Corporation mengimplementasikan arsitektur CISC/RISC ini ke dalam prosesor keluarga-86 dimulai dengan prosesor Pentium, kemudian prosesor P6 atau Pentium Pro (Ryan, 1993 : 84 ; Halfhill, 1995:42 ; Yokota, 1993 : 18-25). Beberapa produsen lain, dengan cara berbeda juga mulai mengadopsi arsitektur campuran CISC/RISC ini misalnya Matsushita Corp dengan prosesor V810, Advanced RISC Machines dengan ARM610, dan Hitachi dengan prosesor SH7032.

Prosesor RISC, yang berkembang dari riset akademis telah menjadi prosesor komersial yang terbukti mampu beroperasi lebih cepat dengan penggunaan luas cip yang efisien. Kemajuan mutakhir yang ditunjukkan oleh mikroprosesor PowerPC 601 dan teknologi emulasi yang antara lain dikembangkan oleh IBM memungkinkan bergesernya dominasi cip-cip keluarga-86 dan kompatibelnya. Bila teknik emulasi terus dikembangkan maka pemakai tidak perlu lagi mempedulikan prosesor apa yang ada di dalam sistem komputernya, selama prosesor tersebut dapat menjalankan sistem operasi ataupun program aplikasi yang diinginkan.

Konsep arsitektur RISC banyak menerapkan proses eksekusi pipeline. Meskipun jumlah perintah tunggal yang diperlukan untuk melakukan pekerjaan yang diberikan mungkin lebih besar, eksekusi secara pipeline memerlukan waktu yang lebih singkat daripada waktu untuk melakukan pekerjaan yang sama dengan menggunakan perintah yang lebih rumit. Mesin RISC memerlukan memori yang lebih besar untuk mengakomodasi program yang lebih besar. IBM 801 adalah prosesor komersial pertama yang menggunakan pendekatan RISC. Lebih lanjut untuk memahami RISC, diawali dengan tinjauan singkat tentang karakteristik eksekusi instruksi.

Aspek komputasi yang ditinjau dalam merancang mesin RISC adalah sebagai berikut:

1. Operasi-operasi yang dilakukan:
2. Hal ini menentukan fungsi-fungsi yang akan dilakukan oleh CPU dan interaksinya dengan memori.
3. Operand-operand yang digunakan:
4. Jenis-jenis operand dan frekuensi pemakaiannya akan menentukan organisasi memori untuk menyimpannya dan mode pengalamatan untuk mengaksesnya.
5. Pengurutan eksekusi:
6. Hal ini akan menentukan kontrol dan organisasi pipeline.

**Ciri-ciri Prosesor RISC**

Sebenarnya, prosesor RISC tidak sekedar memiliki instruksi-instruksi yang sedikit dan sederhana seperti namanya tetapi juga mencakup banyak ciri-ciri lain yang tidak semuanya disepakati oleh kalangan perancang sendiri. Meskipun demikian, banyak yang telah bersepakat bahwa prosesor memiliki ciri-ciri tertentu untuk membedakannya dengan prosesor bukan RISC.

* Pertama, prosesor RISC mengeksekusi instruksi pada setiap satu siklus detak (Robinson, 1987 : 144; Johnson, 1987 : 153). Hasil penelitihan IBM (International Business Machine) menunjukkan bahwa frekuensi penggunaan instruksi-instruksi kompleks hasil kompilasi sangat kecil dibanding dengan instruksi-instruksi sederhana. Dengan perancangan yang baik instruksi sederhana dapat dibuat agar bisa dieksekusi dalam satu siklus detak. Ini tidak berarti bahwa dengan sendirinya prosesor RISC mengeksekusi program secara lebih cepat dibanding prosesor CISC. Analogi sederhananya adalah bahwa kecepatan putar motor(putaran per menit) yang makin tinggi pada kendaraan tidaklah berarti bahwa jarak yang ditempuh kendaraan (meter per menit) tersebut menjadi lebih jauh, karena jarak tempuh masih bergantung pada perbandingan roda gigi yang dipakai.
* Kedua, instruksi pada prosesor RISC memiliki format-tetap, sehingga rangkaian pengontrol instruksi menjadi lebih sederhana dan ini berarti menghemat penggunaan luasan keping semikonduktor. Bila prosesor CISC (misalnya Motorola 68000 atau Zilog Z8000) memanfaatkan 50% - 60% dari luas keping semikonduktor untuk rangkaian pengontrolnya, prosesor RISC hanya memerlukan 6%-10%. Eksekusi instruksi menjadi lebih cepat karena rangkaian menjadi lebih sederhana (Robinson, 1987 : 144; Jonhson 1987 : 153).
* Ketiga, instruksi yang berhubungan dengan memori hanya instruksi isi (load) dan instruksi simpan (store) , instruksi lain dilakukan dalam register internal prosesor. Cara ini menyederhanakan mode pengalamatan(addressing) dan memudahkan pengulangan kembali instruksi untuk kondisi-kondisi khusus yang dikehendaki (Robinson, 1987 : 144; Jonhson, 1987: 153). Dengan ini pula perancang lebih menitikberatkan implementasi lebih banyak register dalam cip prosesor. Dalam prosesor RISC, 100 buah register atau lebih adalah hal yang biasa. Manipulasi data yang terjadi pada register yang umumnya lebih cepat daripada dalam memori menyebabkan prosesor RISC berpotensi beroperasi lebih cepat.
* Keempat, prosesor RISC memerlukan waktu kompilasi yang lebih lama daripada prosesor RISC. Karena sedikitnya pilihan instruksi dan mode pengalamatan yang dimiliki prosesor RISC, maka diperlukan optimalisasi perancangan kompilator agar mampu menyusun urutan instruksi-instruksi sederhana secara efisien dan sesuai dengan bahasa pemrograman yang dipilih. Keterkaitan desain prosesor RISC dengan bahasa pemrograman memungkinkan dirancangnya kompilator yang dioptimasi untuk bahasa target tersebut.

Kesimpulan umum dari ciri-ciri RISC adalah sebagai berikut :

* Instruksi berukuran tunggal
  + - * Ukuran yang umum adalah 4 byte
    - Jumlah pengalamatan data sedikit, biasanya kurang dari 5 buah.
* Tidak terdapat pengalamatan tak langsung yang mengharuskan melakukan sebuah akses memori agar memperoleh alamat operand lainnya dalam memori.
  + Tidak terdapat operasi yang menggabungkan operasi load/store dengan operasi aritmatika, seperti penambahan ke memori dan penambahan dari memori.
  + Tidak terdapat lebih dari satu operand beralamat memori per instruksi
  + Tidak mendukung perataan sembarang bagi data untuk operasi load/ store.
  + Jumlah maksimum pemakaian memori manajemen bagi suatu alamat data adalah sebuah instruksi .
  + Jumlah bit bagi integer register spesifier sama dengan 5 atau lebih, artinya sedikitnya 32 buah register integer dapat direferensikan sekaligus secara eksplisit.
  + Jumlah bit floating point register spesifier sama dengan 4 atau lebih, artinya sedikitnya 16 register floating point dapat direferensikan sekaligus secara eksplisit.

Adapun aspek-aspek komputasinya adalah :

1. Operasi-operasi yang dilakukan ,.
2. Operand-operand yang digunakan,
3. Pengurutan eksekusi,.
4. **Operasi**

Beberapa penelitian telah menganalisis tingkah laku program HLL (High Level Language). Assignment Statement sangat menonjol yang menyatakan bahwa perpindahan sederhana merupakan satu hal yang penting. Hasil penelitian ini merupakan hal yang penting bagi perancang set instruksi mesin yang mengindikasikan jenis instruksi mana yang sering terjadi karena harus didukung optimal.

1. **Operand**

Penelitian Paterson telah memperhatikan [PATT82a] frekuensi dinamik terjadinya kelas-kelas variabel. Hasil yang konsisten diantara program pascal dan C menunjukkan mayoritas referensi menunjuk ke variable scalar. Penelitian ini telah menguji tingkah laku dinamik program HLL yang tidak tergantung pada arsitektur tertentu. Penelitian [LUND77] menguji instruksi DEC-10 dan secara dinamik menemukan setiap instruksi rata-rata mereferensi 0,5 operand dalam memori dan rata-rata mereferensi 1,4 register. Tentu saja angka ini tergantung pada arsitektur dan kompiler namun sudah cukup menjelaskan frekuensi pengaksesan operand sehingga menyatakan pentingnya sebuah arsitektur.

1. **Procedure Calls**

Dalam HLL procedure call dan return merupakan aspek penting karena merupakan operasi yang membutuhkan banyak waktu dalam program yang dikompalasi sehingga banyak berguna untuk memperhatikan cara implementasi opperasi ini secara efisien. Adapun aspeknya yang penting adalah jumlah parameter dan variabel yang berkaitan dengan prosedur dan kedalaman pensarangan (nesting).

**Kelebihan Dan Kekurangan Teknologi RISC**

Teknologi RISC relatif masih baru oleh karena itu tidak ada perdebatan dalam menggunakan RISC ataupun CISC, karena tekhnologi terus berkembang dan arsitektur berada dalam sebuah spektrum, bukannya berada dalam dua kategori yang jelas maka penilaian yang tegas akan sangat kecil kemungkinan untuk terjadi.

**Kelebihan**

1. Berkaitan dengan penyederhanaan kompiler, dimana tugas pembuat kompiler untuk menghasilkan rangkaian instruksi mesin bagi semua pernyataan HLL. Instruksi mesin yang kompleks seringkali sulit digunakan karena kompiler harus menemukan kasus-kasus yang sesuai dengan konsepnya. Pekerjaan mengoptimalkan kode yang dihasilkan untuk meminimalkan ukuran kode, mengurangi hitungan eksekusi instruksi, dan meningkatkan pipelining jauh lebih mudah apabila menggunakan RISC dibanding menggunakan CISC.
2. Arsitektur RISC yang mendasari PowerPC memiliki kecenderungan lebih menekankan pada referensi register dibanding referensi memori, dan referensi register memerlukan bit yang lebih sedikit sehingga memiliki akses eksekusi instruksi lebih cepat.
3. Kecenderungan operasi register ke register akan lebih menyederhanakan set instruksi dan menyederhanakan unit kontrol serta pengoptimasian register akan menyebabkan operand-operand yang sering diakses akan tetap berada dipenyimpan berkecepatan tinggi.
4. Penggunaan mode pengalamatan dan format instruksi yang lebih sederhana.

**Kekurangan**

Kelemahan utama dari RISC ialah jumalh instruksi yang sedikit. Hal ini mengakibatkan untuk melakukan suatu tugas akan dibutuhkan instruksi yang lebih banyak bila dibandingkan CISC. Hasilnya ialah jumlah ukuran program akan lebih besar bila dibandingkan CISC. Penggunaan memori akan semakin meningkat dan lalu lintas instruksi antara CPU dan memori akan meningkat pula.

Prosesor RISC, yang berkembang dari riset akademis telah menjadi prosesor komersial yang terbukti mampu beroperasi lebih cepat dengan penggunaan luas chip yang efisien. Kemajuan mutakhir yang ditunjukkan oleh mikroprosesor PowerPC 601 dan teknologi emulasi yang antara lain dikembangkan oleh IBM memungkinkan bergesernya dominasi chip-chip keluarga-86 dan kompatibelnya. Program yang dihasilkan dalam bahasa simbolik akan lebih panjang (instruksinya lebih banyak).

1. Program berukuran lebih besar sehingga membutuhkan memori yang lebih banyak, ini tentunya kurang menghemat sumber daya.
2. Program yang berukuran lebih besar akan menyebabkan
3. Menurunnya kinerja, yaitu instruksi yang lebih banyak artinya akan lebih banyak byte-byte instruksi yang harus diambil.
4. Pada lingkungan paging akan menyebabkan kemungkinan terjadinya page fault lebih besar

**Analisis Perbedaan Karakteristik RISC dan CISC**

CISC dan RISC perbedaannya tidak signifikan jika hanya dilihat dari terminologi set instruksinya yang kompleks atau tidak (reduced). Lebih dari itu, RISC dan CISC berbeda dalam filosofi arsitekturnya. Filosofi arsitektur CISC adalah memindahkan kerumitan software ke dalam hardware. Teknologi pembuatan IC saat ini memungkinkan untuk menamam ribuan bahkan jutaan transistor di dalam satu dice. Bermacam-macam instruksi yang mendekati bahasa pemrogram tingkat tinggi dapat dibuat dengan tujuan untuk memudahkan programmer membuat programnya.

Beberapa prosesor CISC umumnya memiliki microcode berupa firmware internal di dalam chip-nya yang berguna untuk menterjemahkan instruksi makro. Mekanisme ini bisa memperlambat eksekusi instruksi, namun efektif untuk membuat instruksi-instruksi yang kompleks. Untuk aplikasi-aplikasi tertentu yang membutuhkan singlechip komputer, prosesor CISC bisa menjadi pilihan.

Filosofi arsitektur RISC adalah arsitektur prosesor yang tidak rumit dengan membatasi jumlah instruksi hanya pada instruksi dasar yang diperlukan saja. Kerumitan membuat program dalam bahasa mesin diatasi dengan membuat bahasa program tingkat tinggi dan compiler yang sesuai. Karena tidak rumit, teorinya mikroprosesor RISC adalah mikroprosesor yang low-cost dalam arti yang sebenarnya. Namun demikian, kelebihan ruang pada prosesor RISC dimanfaatkan untuk membuat sistem-sistem tambahan yang ada pada prosesor modern saat ini. Banyak prosesor RISC yang di dalam chip-nya dilengkapi dengan sistem superscalar, pipelining, caches memory, register-register dan sebagainya, yang tujuannya untuk membuat prosesor itu menjadi semakin cepat.

Sudah sering kita mendengar debat yang cukup menarik antara komputer personal IBM dan kompatibelnya yang berlabel Intel Inside dengan komputer Apple yang berlabel PowerPC. Perbedaan utama antara kedua komputer itu ada pada tipe prosesor yang digunakannya. Prosesor PowerPC dari Motorola yang menjadi otak utama komputer Apple Macintosh dipercaya sebagai prosesor RISC, sedangkan Pentium buatan Intel diyakini sebagai prosesor CISC. Kenyataannya komputer personal yang berbasis Intel Pentium saat ini adalah komputer personal yang paling banyak populasinya. Tetapi tidak bisa pungkiri juga bahwa komputer yang berbasis RISC seperti Macintosh, SUN adalah komputer yang handal dengan sistem pipelining, superscalar, operasi floating point dan sebagainya.

Sebenarnya, prosesor RISC tidak sekedar memiliki instruksi-instruksi yang sedikit dan sederhana seperti namanya tetapi juga mencakup banyak ciri-ciri lain yang tidak semuanya disepakati oleh kalangan perancang sendiri. Meskipun demikian, banyak yang telah bersepakat bahwa prosesor memiliki ciri-ciri tertentu untuk membedakannya dengan prosesor RISC dan CISC.

Pertama, prosesor RISC mengeksekusi instruksi pada setiap satu siklus detak. Hasil penelitian IBM (International Business Machine) menunjukkan bahwa frekuensi penggunaan instruksi-instruksi kompleks hasil kompilasinya sangat kecil dibanding dengan instruksi-instruksi sederhana. Dengan perancangan yang baik instruksi sederhana dapat dibuat agar bisa dieksekusi dalam satu siklus detak. Ini tidak berarti bahwa dengan sendirinya prosesor RISC mengeksekusi program secara lebih cepat dibanding prosesor CISC. Analogi sederhananya adalah bahwa kecepatan putar motor (putaran per menit) yang makin tinggi pada kendaraan tidaklah berarti bahwa jarak yang ditempuh kendaraan (meter per menit) tersebut menjadi lebih jauh, karena jarak tempuh masih bergantung pada perbandingan roda gigi yang dipakai.

Kedua, instruksi pada prosesor RISC memiliki format-tetap, sehingga rangkaian pengontrol instruksi menjadi lebih sederhana dan ini berarti menghemat penggunaan luasan keping semikonduktor. Bila prosesor CISC (misalnya Motorola 68000 atau Zilog Z8000) memanfaatkan 50% – 60% dari luas keping semikonduktor untuk rangkaian pengontrolnya, prosesor RISC hanya memerlukan 6%-10%. Eksekusi instruksi menjadi lebih cepat karena rangkaian menjadi lebih sederhana (Robinson, 1987 : 144; Jonhson 1987 : 153).

Ketiga, instruksi yang berhubungan dengan memori hanya instruksi isi (load) dan instruksi simpan (store) , instruksi lain dilakukan dalam register internal prosesor. Cara ini menyederhanakan mode pengalamatan (addressing) dan memudahkan pengulangan kembali instruksi untuk kondisi-kondisi khusus yang dikehendaki. Dengan ini pula perancang lebih menitikberatkan implementasi banyak register dalam chip prosesor. Dalam prosesor RISC, 100 buah register atau lebih adalah hal yang biasa. Manipulasi data yang terjadi pada register yang umumnya lebih cepat daripada dalam memori menyebabkan prosesor RISC berpotensi beroperasi lebih cepat.

Keempat, prosesor RISC memerlukan waktu kompilasi yang lebih lama daripada prosesor CISC. Karena sedikitnya pilihan instruksi dan mode pengalamatan yang dimiliki prosesor RISC, maka diperlukan optimalisasi perancangan kompilator agar mampu menyusun urutan instruksi-instruksi sederhana secara efisien dan sesuai dengan bahasa pemrograman yang dipilih. Keterkaitan desain prosesor RISC dengan bahasa pemrograman memungkinkan dirancangnya kompilator yang dioptimasi untuk bahasa target tersebut.

Kelima, penggunaan prosesor CISC hanya menghemat sedikit atau sama sekali tidak menghemat memori jika dibandingkan RISC. Kemudian compiler-kompiler pada mesin CISC cenderung menggunakan instruksi-instruksi yang lebih sederhana, karena itu keringkasan instruksi kompleks jarang sekali memegang peranan. Selain itu, karena pada CISC terdapat instruksi yang lebih banyak, maka diperlukan code yang lebih panjang, yang akan menghasilkan instruksi yang lebih panjang pula. RISC cenderung menekankan pada referensi register dibandingkan pada referensi memori, dan referensi register memerlukan bit yang jumlahnnya lebih sedikit.

**Perbedaan RISC dengan CISC dilihat dari segi instruksinya**

**RISC ( Reduced Instruction Set Computer )**

* Menekankan pada perangkat lunak, dengan sedikit transistor
* Instruksi sederhana bahkan single
* Load / Store atau memory ke memory bekerja terpisah
* Ukuran kode besar dan kecapatan lebih tinggi
* Transistor didalamnya lebih untuk meregister memori

**CISC ( Complex Instruction Set Computer )**

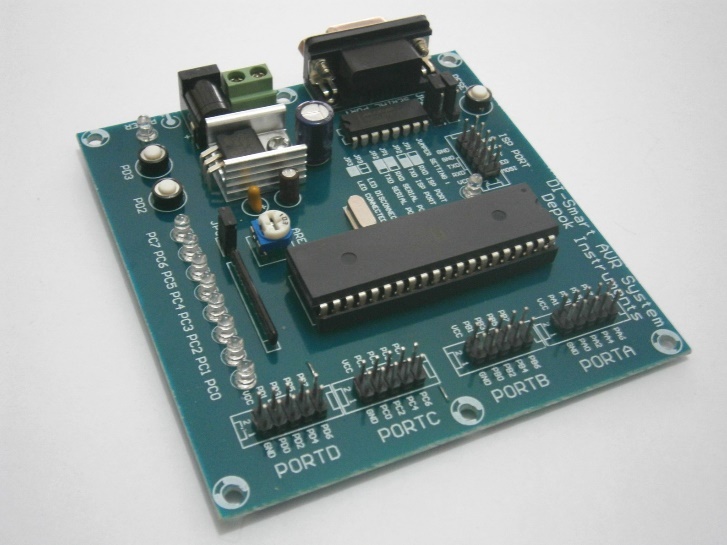
* Lebih menekankan pada perangkat keras, sesuai dengan takdirnya untuk pragramer.
* Memiliki instruksi komplek. Load / Store atau Memori ke Memori bekerjasama
* Memiliki ukuran kode yang kecil dan kecepatan yang rendah.
* Transistor di dalamnya digunakan untuk menyimpan instruksi – instruksi bersifat komplek

**Perbedaan cisc dan risc berdasarkan karakteristiknya**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Karakteristik** | **CISC** | | | **RISC** | |
| **IBM**  **370/168** | **Vax**  **11/780** | **Intel**  **80486** | **Motorola**  **88000** | **MIPS**  **R4000** |
| Tahun Dibuat | 1973 | 1978 | 7989 | 1988 | 7991 |
| Jumlah Intruksi | 208 | 303 | 235 | 51 | 94 |
| Intruksi Bytes | 2-6 | 2-57 | 1-11 | 4 | 32 |
| Mode Pengamatan | 4 | 22 | 11 | 3 | 1 |
| Jumlah register  General-Purpose | 16 | 16 | 8 | 32 | 32 |
| Ukuran Memori  Kontrol (Kbits) | 420 | 486 | 246 | - | - |
| Ukuran Cache (Kbytes) | 64 | 64 | 8 | 16 | 128 |

**Jenis- Jenis Mikrokontroler**

1. **Mikrokontroler AVR**

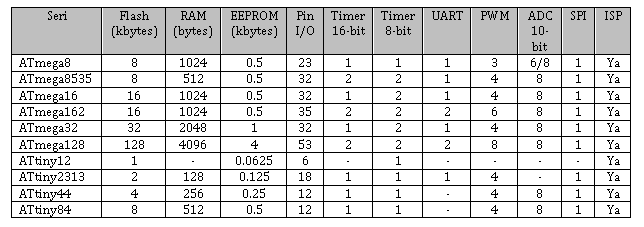


Mikrokonktroler Alv and Vegard’s Risc processor atau sering disingkat AVR merupakan mikrokonktroler RISC 8 bit. Karena RISC inilah sebagian besar kode instruksinya dikemas dalam satu siklus clock. Mikrokontroler AVR merupakan salah satu jenis arsitektur mikrokontroler yang menjadi andalan Atmel. Arsitektur ini dirancang memiliki berbagai kelebihan dan merupakan penyempurnaan dari arsitektur mikrokontroler-mikrokontroler yang sudah ada.

Berbagai seri mikrokontroler AVR telah diproduksi oleh Atmel dan digunakan di dunia sebagai mikrokontroler yang bersifat low cost dan high performance. Di Indonesia, mikrokontroler AVR banyak dipakai karena fiturnya yang cukup lengkap, mudah untuk didapatkan, dan harganya yang relatif terjangkau.

**Varian Mikrokontroler AVR**

Antar seri mikrokontroler AVR memiliki beragam tipe dan fasilitas, namun kesemuanya memiliki arsitektur yang sama, dan juga set instruksi yang relatif tidak berbeda. Tabel dibawah ini membandingkan beberapa seri mikrokontroler AVR buatan Atmel.

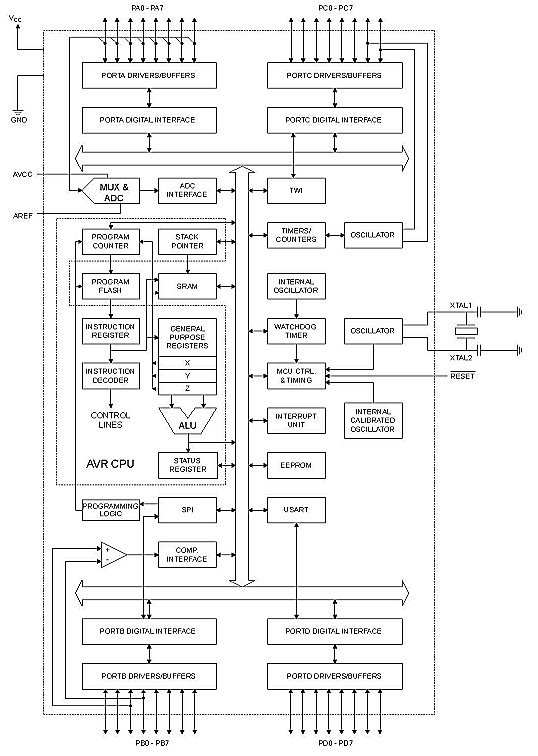


Keterangan:

* Flash adalah suatu jenis Read Only Memory yang biasanya diisi dengan program hasil buatan manusia yang harus dijalankan oleh mikrokontroler
* RAM (Random Acces Memory) merupakan memori yang membantu CPU untuk penyimpanan data sementara dan pengolahan data ketika program sedang running
* EEPROM (Electrically Erasable Programmable Read Only Memory) adalah memori untuk penyimpanan data secara permanen oleh program yang sedang running
* Port I/O adalah kaki untuk jalur keluar atau masuk sinyal sebagai hasil keluaran ataupun masukan bagi program
* Timer adalah modul dalam hardware yang bekerja untuk menghitung waktu/pulsa
* UART (Universal Asynchronous Receive Transmit) adalah jalur komunikasi data khusus secara serial asynchronous
* PWM (Pulse Width Modulation) adalah fasilitas untuk membuat modulasi pulsa
* ADC (Analog to Digital Converter) adalah fasilitas untuk dapat menerima sinyal analog dalam range tertentu untuk kemudian dikonversi menjadi suatu nilai digital dalam range tertentu
* SPI (Serial Peripheral Interface) adalah jalur komunikasi data khusus secara serial secara serial synchronous
* ISP (In System Programming) adalah kemampuan khusus mikrokontroler untuk dapat diprogram langsung dalam sistem rangkaiannya dengan membutuhkan jumlah pin yang minimal

**Arsitektur Mikrokontroler AVR**

Mikrokontroler AVR sudah menggunakan konsep arsitektur Harvard yang memisahkan memori dan bus untuk data dan program, serta sudah menerapkan single level pipelining. Selain itu mikrokontroler AVR juga mengimplementasikan RISC (Reduced Instruction Set Computing) sehingga eksekusi instruksi dapat berlangsung sangat cepat dan efisien



*Blok Diagram Mikrokontroler AVR*

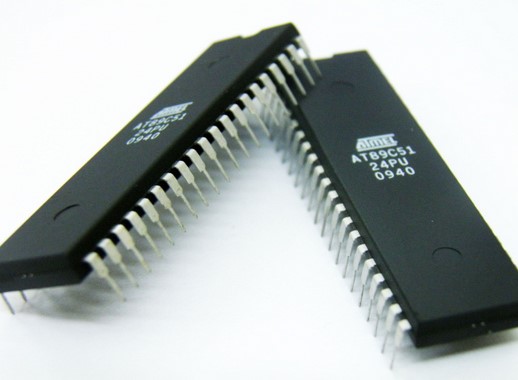
Salah satu seri mikrokontroler AVR yang banyak menjadi andalan saat ini adalah tipe ATtiny2313 dan ATmega8535. Seri ATtiny2313 banyak digunakan untuk sistem yang relatif sederhana dan berukuran kecil. Berikut adalah feature-feature mikrokontroler seri ATtiny2313.

* Kapasitas memori Flash 2 Kbytes untuk program
* Kapasitas memori EEPROM 128 bytes untuk data
* Maksimal 18 pin I/O
* 8 interrupt
* 8-bit timer
* Analog komparator
* On-chip oscillator
* Fasilitas In System Programming (ISP)

Sedangkan ATmega8535 banyak digunakan untuk sistem yang kompleks, memiliki input sinyal analog, dan membutuhkan memori yang relatif lebih besar. Berikut adalah feature-feature mikrokontroler seri ATmega8535.

* Memori Flash 8 Kbytes untuk program
* Memori EEPROM 512 bytes untuk data
* Memori SRAM 512 bytes untuk data
* Maksimal 32 pin I/O
* 20 interrupt
* Satu 16-bit timer dan dua 8-bit timer
* 8 channel ADC 10 bit
* Komunikasi serial melalui SPI dan USART
* Analog komparator
* I/O PWM
* Fasilitas In System Programming (ISP)

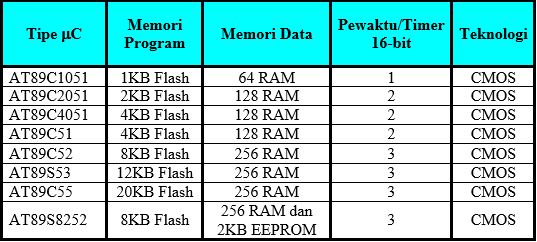
1. **Mikrokontroler MCS-51**



Mikrokonktroler ini termasuk dalam keluarga mikrokonktroler CISC (Complex Instruction Set Computer). Sebagian besar instruksinya dieksekusi dalam 12 siklus clock.

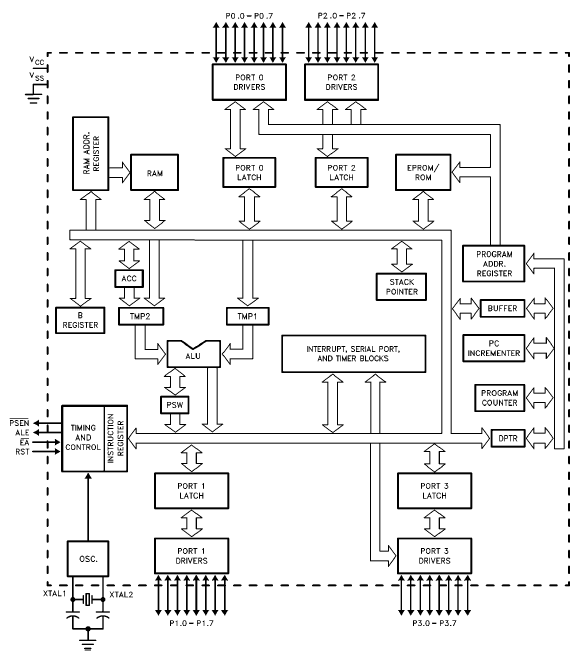
Mikrokontroler MCS51 buatan Atmel terdiri dari dua versi, yaitu versi 20 kaki dan versi 40 kaki. Semua mikrokontroler ini dilengkapi dengan Flash PEROM (Programmable Eraseable Read Only Memory) sebagai media memori-program, dan susunan kaki IC-IC tersebut sama pada tiap versinya.

Perbedaan dari mikrokontroler-mikrokontroler tersebut terutama terletak pada kapasitas memori-program, memori-data dan jumlah pewaktu 16-bit.

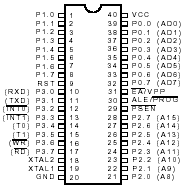


*Perbandingan antar Mikrokontroler MCS51Atmel*

Mikrokontroler MCS51 Atmel versi mini (20 pin) dan versi 40 pin secara garis besar memiliki struktur dasar penyusun arsitektur mikrokontroler yang sama. Bagian-bagian tersebut secara lebih lengkap (detil) ditunjukan dalam diagram blok berikut.



Mikrokontroler MCS51 Atmel versi 40 kaki mempunyai 32 kaki sebagai port paralel dan 8 pin yang lain untuk konfigurasi kerja mikrokontroler. Satu port paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel yang masing-masing dikenal sebagai port 0, port 1, port 2, port 3. Nomor dari masing-masing jalur (kaki) dari port paralel mikrokontroler MCS51 Atmel mulai dari 0 sampai 7, jalur (kaki) pertama dari port 0 disebut sebagai P0.0 dan jalur terakhir untuk port 3 adalah P3.7. Mikrokontroler MCS51 Atmel versi mini mempunyai 20 kaki, 15 kaki diantaranya adalah kaki port 1 dan port 3. 5 kaki yang lain untuk konfigurasi kerja mikrokontroler. Port 1 terdiri dari 8 jalur yaitu P1.0 sampai P1.7 dan port 3 terdiri dari 7 jalur yaitu P3.0 sampai P3.5 dan P3.7. Susunan kaki mikrokontroler MCS51 atmel versi 40 kaki dapat dilihat pada berikut.



**Keterangan**

Fungsi-Fungsi Kaki (Pin):

1. **VCC**

Kaki VCC digunakan untuk masukan suplai tegangan.

1. **GND**

Kaki (pin) GND funsinya sebagai saluran ground atau pentanahan.

1. **RST**

Kaki RST fungsinya sebagai masukan reset. Kondisi “1” selama 2 siklus mesin pada saat oscillator bekerja akan me-reset mikrokontroler yang bersangkutan.

1. **ALE/**

Kaki ALE digunakan sebagai keluaran ALE atau Adreess Latch Enable yang akan menghasilkan pulsa-pulsa untuk menahan byte rendah (low byte) alamat selama mengakses memori eksternal. Kaki ini juga berfungsi sebagai masukan pulsa program (the program pulse input) atau selama pemrograman flash. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan (timing) atau pendekatan (clocking) rangkainan eksternal.

Kaki (Program Store Enable) merupakan sinyal baca untuk memori program eksternal. Saat mikrokontroler MCS51 menjalankan program dari memori eksternal, akan diaktifkan dua kali per-siklus mesin, kecuali dua aktivasi dilompati (diabaikan) saat mengakses memori data eksternal.

1. **VPP**

Kaki /VPP ( Exkternal Access Enable) fungsinya sebagai kontrol untuk mengakses memori. harus dihubungkan ke ground, jika mikrokontroler akan mengeksekusi program dari memori eksrternal. Selain itu harus dihubungkan ke VCC jika akan mengakses program secara internal. Kaki ini juga berfungsi untuk menerima tegangan 12V (VPP) selama pemrograman flash, khususnya untuk tipe mikrokontroler 12V volt.

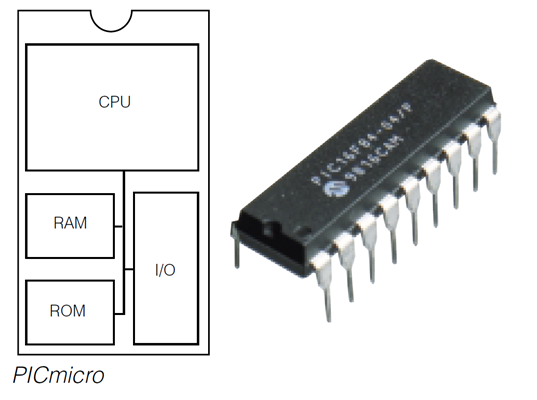
1. **XTAL1**

Kaki XTAL1 merupakan masukan untuk penguat inverting oscillator dan masukan untuk clock internal pada rangkaian operasi mikrokontroler.

1. **XTAL2**

Kaki XTAL2 merupakan keluaran dari rangkaian penguat inverting oscilator

**Mikrokontroler PCI**

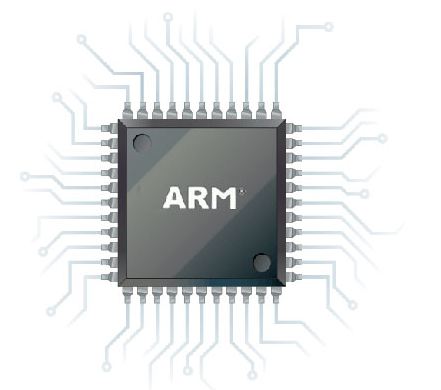


Pada awalnya, PIC merupakan kependekan dari Programmable Interface Controller. PIC termasuk keluarga mikrokonktroler berarsitektur Harvard yang dibuat oleh Microchip Technology. Awalnya dikembangkan oleh Divisi Mikroelektronik General Instruments dengan nama PIC1640.

PIC memungkinkan Anda untuk mengontrol perangkat output ketika mereka dipicu oleh sensor dan switch. Program dapat dihasilkan dengan menggunakan diagram alur dalam perangkat lunak komputer, yang kemudian dapat di-download ke dalam chip PIC. Mereka dapat ditulis ulang sebanyak yang Anda inginkan. Memori jenis ini disebut memori flash.

Sebuah mikrokontroler PIC adalah sirkuit terpadu tunggal cukup kecil untuk muat di telapak tangan dan berisi memori pengolahan unit, Jam dan sirkuit Input / Output dalam satu unit. Sebuah mikrokontroler PIC, oleh karena itu, sering digambarkan sebagai komputer dalam sirkuit terpadu. Mikrokontroler PIC dapat dibeli kosong dan kemudian diprogram dengan program kontrol tertentu. Mikrokontroler PIC juga dapat dibeli dengan pra–diprogram seperangkat perintah yang memungkinkan download langsung dari kabel komputer dan mengurangi biaya peralatan pemrograman.

**Mikrokontroler ARM**



ARM adalah prosesor dengan arsitektur set instruksi 32­bit RISC (Reduced Instruction Set Computer) yang dikembangkan oleh ARM Holdings. ARM merupakan singkatan dari Advanced RISC Machine (sebelumnya lebih dikenal dengan kepanjangan Acorn RISC Machine). Pada awalnya ARM prosesor dikembangkan untuk PC (Personal Computer) oleh Acorn Computers, sebelum dominasi Intel x86 prosesor­ Microsoft di IBM PC kompatibel menyebabkan Acorn Computers bangkrut.

Melalui izin dari seluruh dunia, arsitektur ARM adalah yang paling umum dilaksanakan 32-bit set instruksi arsitektur. Arsitektur ARM diimplementasikan pada Windows, Unix, dan sistem operasi mirip Unix, termasuk Apple iOS, Android, BSD, Inferno, Solaris, WebOS, Plan 9 dan GNU / Linux. Advanced RISC Machine awalnya dikenal sebagai Mesin Acorn RISC.

**Mikrokontroler AT89S52**



Mikrokontroler 89S52 merupakan versi terbaru dibandingkan mikrokontroler AT89C51 yang telah banyak digunakan saat ini. AT89S52 mmpunyai kelebihan yaitu mempunyai flash memori sebesar 8K bytei, RAM 256 byte serta 2 buah data pointer 16 bit, Spesifikasinya:

* Kompatibel dengan keluarga mikrokontroler MCS51 sebelumnya.
* 8 K Bytes In system Programmable (ISP) flash memori dengan kemampuan 1000 kali baca/tulis
* Tegangan kerja 4-5 V
* Bekerja dengan rentang 0 – 33MHz
* 256×8 bit RAM internal
* 32 jalur I/O dapat deprogram
* buah 16 bit Timer/Counter
* 8 sumber interrupt
* saluran full dupleks serial UART
* watchdog timer
* dual data pointer

Mode pemrograman ISP yang fleksibel (Byte dan Page Mode) Jenis-jenis Mikrokontroler Atmel lain yang ada di pasaran adalah sebagai berikut:

**Atmel AT91 series (ARM THUMB architecture)**

* Atmel AVR32
* AT90, Tiny & Mega series – AVR (Atmel Norway design)
* Atmel AT89 series (Intel 8051/MCS51 architecture)
* MARC4
* AMCC

Hingga Mei 2004, mikrokontroler ini masih dikembangkan dan dipasarkan oleh IBM, hingga kemudian keluarga 4xx dijual ke Applied Micro Circuits Corporation, jenis-jenisnya yaitu:

* 403 PowerPC CPU (PPC 403GCX)
* 405 PowerPC CPU (PPC 405EP, PPC 405GP/CR, PPC 405GPr, PPC NPe405H/L)
* 440 PowerPC Book-E CPU (PPC 440GP, PPC 440GX, PPC 440EP/EPx/GRx, PPC 440SP/SPe)

**Cypress MicroSystem**

Jenis dari Cypress MicroSystems yang ada di pasaran adalah CY8C2xxxx (PSoC)

**Freescale Semiconductor**

Hingga 2004, mikrokontroler ini dikembangkan dan dipasarkan oleh Motorola, yang divisi semikonduktornya dilepas untuk mempermudah pengembangan Freescale Semiconductor, adapun jenis-jenisnya yaitu sebagai berikut:

* 8-bit (68HC05 (CPU05), 68HC08 (CPU08), 68HC11 (CPU11))
* 16-bit (68HC12 (CPU12), 68HC16 (CPU16), Freescale DSP56800 (DSPcontroller))
* 32-bit (Freescale 683XX (CPU32), MPC500, MPC 860 (PowerQUICC), MPC 8240/8250 (PowerQUICC II), MPC 8540/8555/8560 (PowerQUICC III))

**Fujitsu**

Jenis chip mikrokontroler yang dikeluarkan oleh fujitsu diantaranya adalah sebagai berikut:

* F²MC Family (8/16 bit)
* FR Family (32 bit)
* FR-V Family (32 bit RISC)

**Holtek**

Chip mikrokontroler keluaran holtek adalah jenis HT8.

**Intel**

Intel adalah salah satu perusahan yang banyak mengeluarkan jenis chip di pasaran, secara umum intel mengeluarkan dua jenis chip mikrokontroler yaitu:

* 8-bit (8XC42, MCS48, MCS51, 8061, 8xC251)
* 16-bit (80186/88, MCS96, MXS296, 32-bit, 386EX, i960)

**Microchip**

Dalam mengeluarkan prduknya, microchip membagi produknya kedalam beberapa jenis yaitu:

* Low End, Mikrokontroler PIC 12-bit
* Mid Range, Mikrokontroler PIC 14-bit (PIC16F84, PIC16F877)
* 16-bit instruction PIC
* High End, Mikrokontroler PIC 16-bit

**National Semiconductor**

Jenis chip mikrokontroler yang dikeluarkan oleh National

Semiconductor adalah jenis COP8 dan CR16.

**NEC**

NEC mempunyai beberapa jenis chip mikrokontroler yang ada dipasaran yaitu : jenis 17K, 75X, 78K, V850.

**Philips Semiconductors**

Ada tiga jenis chip mikrokontroler yang dikeluarkan oleh perusahaan ini yaitu : LPC2000, LPC900, LPC700.

**Renesas Tech. Corp.**

Renesas adalah perusahan patungan Hitachi dan Mitsubishi. Perusahaan ini mengeluarkan beberapa jenis mikrokontroler yang ada dipasaran yaitu : H8, SH, M16C, M32R.

**ST Microelectronics**

STMicroelectronic merupakan salah satu perusahaan yang bergerak juga dalam produksi chip mikrokontroler, diantaranya produknya adalah : ST 62, ST 7.

**Texas Instruments**

Dua jenis chip mikrokontroler yang di produksi oleh perusahaan ini adalah : TMS370, MSP430.

**Western Design Center**

Perusahaan Wistern Design Center memproduksi dua tipe chip mikrokontroler yang beredar di pasaran yaitu:

* Tipe 8-bit (W65C02-based µCs)
* Tipe 16-bit (W65816-based µCs)

**Ubicom**

Ubicom memproduksi beberapa tipe chip mikrokontroler diantaranya adalah:

* SX-28, SX-48, SX-54

Seri Ubicom’s SX series adalah jenis mikrokontroler 8 bit yang, tidak seperti biasanya, memiliki kecepatan tinggi, memiliki sumber daya memori yang besar, dan fleksibilitas tinggi. Beberapa pengguna menganjurkan mikrokontroller pemercepat PICs. Meskipun keragaman jenis mikrokontroler Ubicom’s SX sebenarnya terbatas, kecepatan dan kelebihan sumber dayanya yang besar membuat programmer bisa membuat perangkat virtual lain yang dibutuhkan. Referensi bisa ditemukan di Parallax’s Web site, sebagai penyalur utama.

* IP2022

Ubicom’s IP2022 adalah mikrokontroler 8 bit berkecepatan tinggi (120 MIPs). Fasilitasnya berupa: 64k FLASH code memory, 16k PRAM (fast code dan packet buffering), 4k data memory, 8-channel A/D, various timers, and on-chip support for Ethernet, USB, UART, SPI and

GPSI interfaces.

**Xilinx**

Ada dua jenis chip mikrokontroler yang dikeluarkan oleh perusahaan Xilink diataranya adalah:

* Microblaze softcore 32 bit microcontroller
* Picoblaze softcore 8 bit microcontroller

**ZiLOG**

Dua jenis chip mikrokontroler dari ZiLOG yang ada di pasaran adalah:

* Z8
* Z86E02

Disamping itu, Ada banyak mikrokontroller yang dirancang oleh produsen sebagai sarana hobi. Biasanya mikrokontroller seperti ini dimuati interpreter BASIC, dihubungkan ke bagian Dual Inline Pin bersama power regulator dan beberapa fasilitas lain. PICs sepertinya sangat popular untuk jenis ini, barangkali karena adanya perlindungan terhadap listrik statis. Diantara produk ini adalah:

**Parallax, Inc**

BASIC Stamp. Nama besar di mikrokontroler BASIC, meskipun sebenarnya lamban dan harganya tidak sebanding. SX-Key. Harga murahnya harus dibayar dengan kualitas yang buruk.

**PicAxe**

Murah, tidak lebih dari sekedar PIC yang dimuati BASIC. Bagian programmernya ditancapi dengan 3 resistors. Penawaran BASIC menawarkan fungsionalitas yang besar dengan adanya fasilitas IF..GOTO secara terbatas.

**Kelebihan dan Kekurangan Mikrokontroler**

**Kelebihan:**

1. Bahasa pemrograman assembly sebagai penggerak pada mikrokontroler

Yang mana berpatokan pada digital dasar yang menggunakan bahasa pemrograman assembly, hal ini mengakibatkan melakukan operasi sistem menjadi ringan dan cepat untuk dikerjakan sesuai dengan menggunakan konsep sistem logika. Keuntungan lainnya bahasa assembly dapat sangat mudah dipahami karena parameter untuk input dan output secara langsung yang digunakan bisa diakses tanpa menggunakan perintah yang ribet.

* Tidak menggunakan banyak aturan dalam penulisan bahasa pemrograman yang dipakai dalam desain bahasa pemrograman assembly. Contohnya seperti huruf besar atau kecil dalam menggunakan bahasa assembly.
* Mikrokontroler yang tersusun dalam sebuah cip.
* , I/O, dan memori melakukan integrasi yang menjadi suatu kesatuan sistem yang dapat membuat mikrokontroler bekerja secara kreatif dan inovatif yang bisa mengadaptasikan kebutuhan dari sebuah sistem.
* Sistem running mikrokontroller berdiri sendiri

Sistem running mikrokontroller tidak terkait dengan komputer tersebut sedangkan parameter yang dimiliki komputer hanya dapat dipergunakan mendownload perintah instruksi. Langkah untuk melakukan download pada komputer dengan mikrokontroler sangatlah mudah untuk digunakan karena hanya sedikit saja perintahnya.

1. Mikrokontroler memiliki fasilitas tambahan yang nantinya akan dipergunakan untuk melakukan pengembangan yaitu memori dan I/O inovatif yang akan disesuaikan dengan sistem.
2. Dengan memory yang bersifat nonvolatile yang memungkinkan ic dapat di program ulang. Sehingga jika kita ingin menggunakan ic yang didalamnya sudah terdapat program yang sebelumnya atau kita melakukan kesalahan atas perintah program yang sudah terisi sebelumnya, kita dapat memprogram ulang dengan cara meng-flash tanpa harus membeli ic mikrokontroler yang baru.

* Sistem running microcontroller berdiri sendiri tanpa tergantung dengan komputer sedangkan parameter komputer hanya digunakan untuk download perintah instruksi atau program. Langkah-langkah untuk download komputer dengan mikrokontroler sangat mudah digunakan karena tidak menggunakan banyak perintah.
* Pada mikrokontroler tersedia fasilitas tambahan untuk pengembangan memori dan I/O yang disesuaikan dengan kebutuhan sistem.
* Harga microcontroller lebih murah dan mudah didapat.

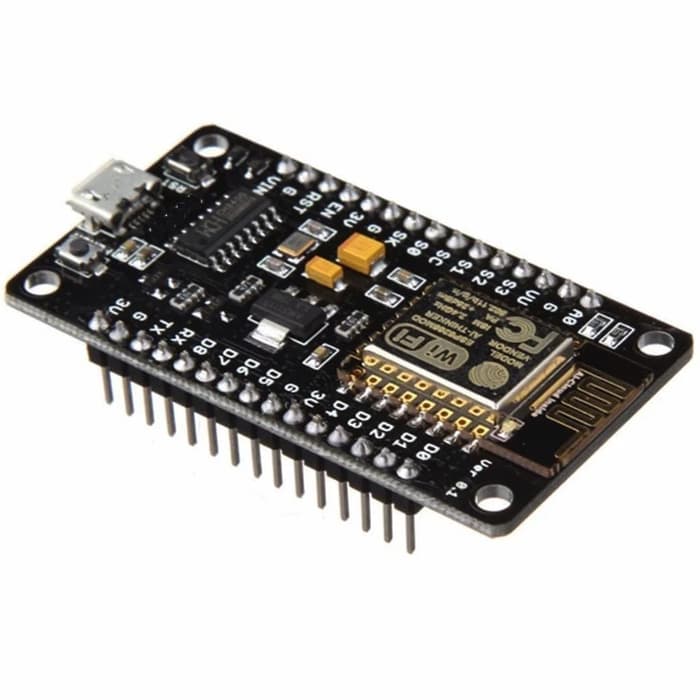
**Kekurangan**

* Kekurangan dari mikrokontroler ialah hanya dapat men konfigurasi dan menjalankan satu perintah saja dalam satu waktu akibatnya perintah selanjutnya harus menunggu hingga perintah yang pertama telah selesai dijalankan.
* proses yang dapat dijalankan pada mikrokontroler tidak data melakukan berbagai proses dalam waktu yang bersamaan. Jadi mikrokontroler hanya dapat menjalankan satu perintah atau instruksi dalam satu waktu sehingga perinta atau instruksi yang lain harus menunggu hingga instruksi yang pertama selesai dijalankan.

**Memilih Mikrokontroler untuk Sistem IoT kita**

Terdapat banyak jenis development board dan mikrokontroler tersedia dari berbagai perusahaan seperti: TI, Samsung, Arduino, Raspberry Pi dan lainnya. Memilih yang mana yang cocok dengan kita bergantung pada beberapa faktor yang juga sangat beragam, tergantung dari sifat aplikasi yang akan kita kembangkan:

* **Kompatibilitas**: Apakah MCUnya mendukung sensor dan aktuator yang ingin kita gunakan? Berdasarkan sensor dan aktuator, kita mungkin perlu banyak atau hanya beberapa port/pin GPIO. Kita mesti memastikan apakah kita punya cukup port yang tersedia.
* **Arsitektur**: Apakah arsitekturnya cukup canggih untuk menangani kompleksitas dari program kita? Kebanyakan aplikasi menggunakan AVR, ARM, MIPS, atau x86. Memilih salah satunya bergantung pada kebutuhan funsional dari aplikasi kita dan seberapa banyak daya komputasi yang diperlukan oleh sistem kita.
* **Memori:** Apakah MCUnya memiliki cukup memori - RAM dan Flash - untuk program kita? Sangat direkomendasikan untuk memilih MCU yang memiliki jumlah memori ekstra untuk pembaharuan dimasa yang akan datang. Ini akan banyak menghemat waktu, biaya, dan menghindari kepala pusing dalam perjalanan yang cukup panjang!
* **Ketersediaan**: Apakah kita bisa mendapatkan MCUnya dengan mudah, dalam jumlah yang kita perlukan? Ini adalah faktor yang sangat penting untuk dipertimbangkan disaat awal proses pengembangan, khususnya apabila kita berencana untuk mengembangkan sistem kita lebih besar lagi nanti.
* **Daya**: Sebarapa besar daya yang diperlukan MCUnya? Apakah ia nantinya harus dinyalakan dari sumber daya kabel atau kita bisa menggunakan baterai? Efisiensi daya sangatlah penting untuk dipertimbangkan untuk aplikasi IoT industri karena kita tentunya ingin meminimalisir keperluan untuk mengirim petugas pemeliharaan untuk menginspeksi infrastruktur yang letaknya jauh di pinggiran.
* **Biaya:** Seberapa banyakah setiap unit alatnya menghabiskan biaya? Apakah harganya cocok dengan nilai yang akan dihasilkan? Sekali lagi, kita perlu memikirkan tentang mengembangkan projeknya lebih jauh lagi nanti. Pastikan bahwa anggaran IoT kita mendukung termasuk untuk pengadaaan lebih banyak lagi MCU yang kita pilih.
* **Dukungan pengembangan**: Apakah dokumentasi yang berisi tata cara untuk berbagai aspek dan fitur dari MCU yang kita pilih tersedia dengan baik? Bagaimanakan kondisi komunitas yang menggunakan board ini? Faktor-faktor tersebut sangat krusial untuk dapat mengambil keputusan dan menggunakan MCU kita dengan sebagaimana mestinya. Komunitas online yang bagus dapat membantu membimbing kita saat kita menghadapi kebuntuan atau mendapatkan masalah dengan implementasi kita.
  + - 1. **NodeMCU**



*Gambar 2.3 NodeMCU*

NodeMCU adalah sebuah platform IoT yang bersifat opensource. Terdiri dari perangkat keras berupa *System On Chip* ESP8266. dari ESP8266 buatan Espressif System, juga firmware yang digunakan, yang menggunakan bahasa pemrograman scripting Lua. [Sumardi, 2016] Istilah NodeMCU secara default sebenarnya mengacu pada firmware yang digunakan dari pada perangkat keras development kit NodeMCU bisa dianalogikan sebagai board arduino-nya ESP8266.

NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur – fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip dengan javascript. [14] Selain dapat diprogram menggunakan bahasa LUA dapat juga diprogram menggunakan bahasa C menggunakan arduino IDE. Karena jantung dari NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur – fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip dengan javascript. Beberapa fitur tersebut antara :

1. 0 Port GPIO dari D0 – D10
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 Wire
5. ADC

Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan GPIO dimana jumlah pin bergantung dengan jenis ESP8266 yang kita gunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler

**Sejarah NodeMCU**

Sejarah lahirnya NodeMCU berdekatan dengan rilis ESP8266 pada 30 Desember 2013, Espressif Systems selaku pembuat ESP8266 memulai produksi ESP8266 yang merupakan SoC Wi-Fi yang terintegrasi dengan prosesor Tensilica Xtensa LX106. Sedangkan NodeMCU dimulai pada 13 Oktober 2014 saat Hong me-commit file pertama nodemcu-firmware ke Github. Dua bulan kemudian project tersebut dikembangkan ke platform perangkat keras ketika Huang R meng-commit file dari board ESP8266, yang diberi nama devkit v.0.9.

Berikutnya, di bulan yang sama. Tuan PM memporting pustaka client MQTT dari Contiki ke platform SOC ESP8266 dan di-c0mmit ke project NodeMCU yang membuatnya mendukung protokol IOT MQTT melalui Lua. Pemutakhiran penting berikutnya terjadi pada 30 Januari 2015 ketika Devsaurus memporting u8glib ke project NodeMCU yang memungkinkan NodeMCU bisa mendrive display LCD, OLED, hingga VGA. Demikianlah, project NodeMCU terus berkebang hingga kini berkat komunitas open source dibaliknya, pada musim panas

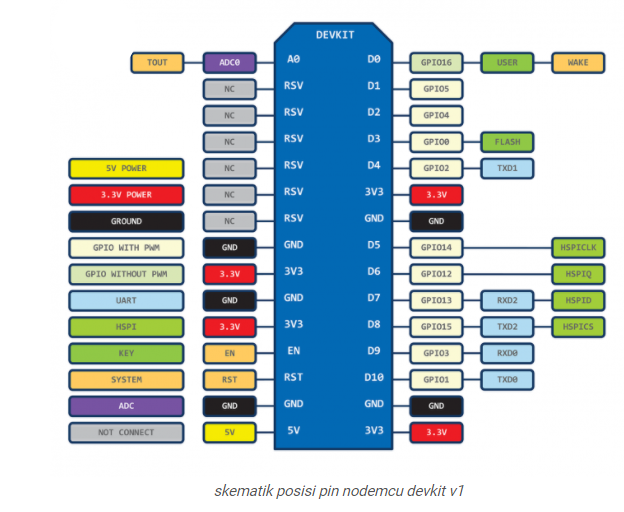
**Versi NodeMCU**

Beberapa pengguna awal masih cukup bingung dengan beberapa kehadiran board NodeMCU. Karena sifatnya yang open source tentu akan banyak produsen yang memproduksinya dan mengembangkannya. Secara umum ada tiga produsen NodeMCU yang produknya kini beredar di pasaran: Amica, DOIT, dan Lolin/WeMos. Dengan beberapa varian board yang diproduksi yakni V1, V2 dan V3.

1. **Generasi pertama / board v.0.9 (Biasa disebut V1)**

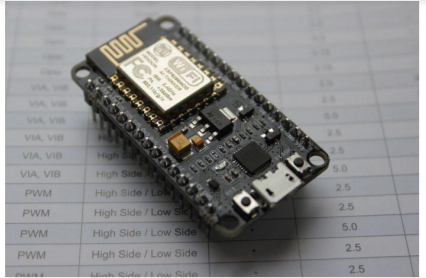


Board versi 0.9 sering disebut di pasar sebagai V.1 adalah versi asli yang berdimensi 47mm x 31mm. Memiliki inti ESP-12 dengan flash memory berukuran 4MB. Berikut adalah pinout dari board v.0.9

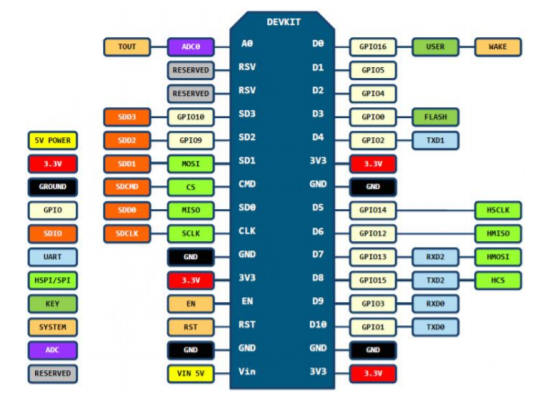


Namun beberapa produk juga ada yang menggunakan chip ESP-12E sebagai inti dari board v.0.9 dengan tampilan board berubah menjadi hitam.

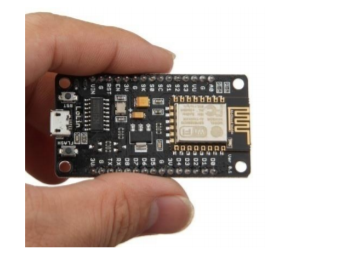
1. **Generasi kedua / board v 1.0 (biasa disebut V2)**



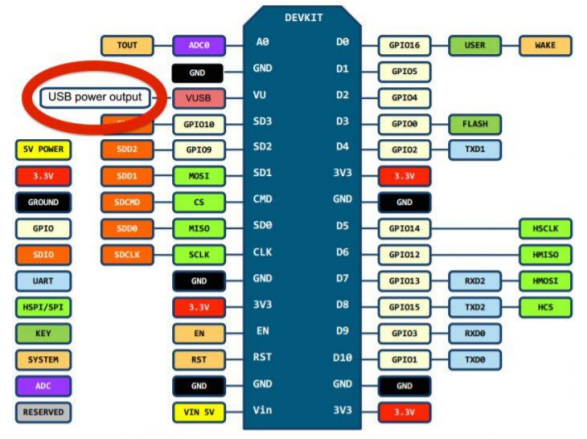
Generasi kedua adalah pengembangan dari versi sebelumnya, dengan chip yang ditingkatkan dari sebelumnya ESP12 menjadi ESP12E. Dan IC Serial diubah dari CHG340 menjadi CP2102



1. **Generasi ketiga / board v 1.0 ( biasa disebut V3 Lolin)**



Sedangkan untuk V3 sebenarnya bukanlah versi resmi yang dirilis oleh NodeMCU. Setidaknya sampai posting ini dibuat, belum ada versi resmi untuk V3 NodeMCU. V3 hanyalah versi yang diciptakan oleh produsen LoLin dengan perbaikan minor terhadap V2. Diklaim memiliki antarmuka USB yang lebih cepat.



Jika anda bandingkan dengan versi sebelumnya, dimensi dari board V3. Akan lebih besar dibanding V2. Lolin menggunakan 2 pin cadangan untuk daya USB dan yang lain untuk GND tambahan. Tentu 3 jenis versi ini akan berkembang dan bertambah seiring dengan waktu karena sifatnya yang opensource. Mungkin beberapa bulan atau beberapa tahun setelah tulisan ini dibuat akan muncul versi- versi lain yang beredar.

**Kelebihan dan Kekurangan dari NodeMCU**

**Kelebihan:**

* + Berbiaya rendah
  + Dukungan terintegrasi untuk jaringan WiFi
  + Ukuran board yang lebih kecil
  + Konsumsi energi yang lebih rendah

**Kekurangan**

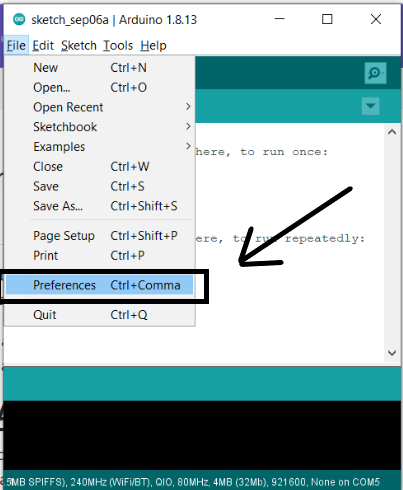
* Perlu mempelajari bahasa baru dan IDE baru
* Pinout yang lebih sedikit
* Dokumentasi yang masih sedikit langka

**Cara Menambahkan NodeMCU pada Arduino IDE**

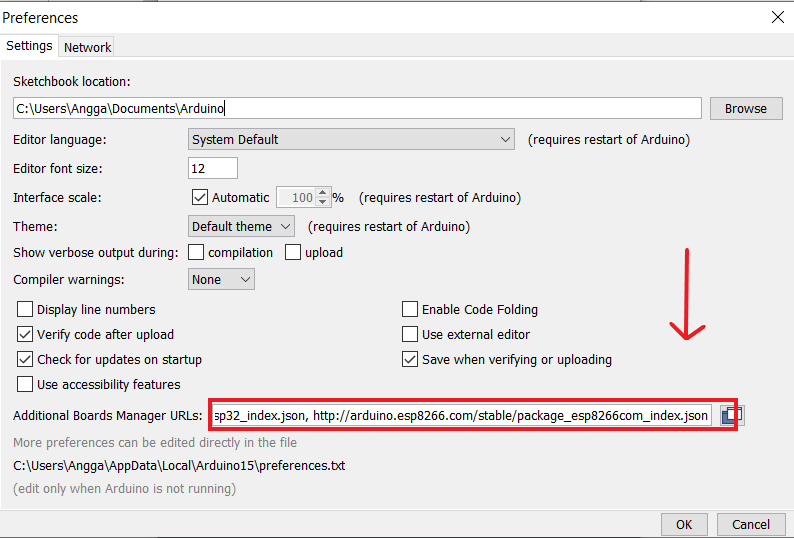
1. Install terlebih dahulu Arduino IDE pada komputer atau laptop anda kemudian jalankan aplikasi tersebut tampilannya seperti berikut:



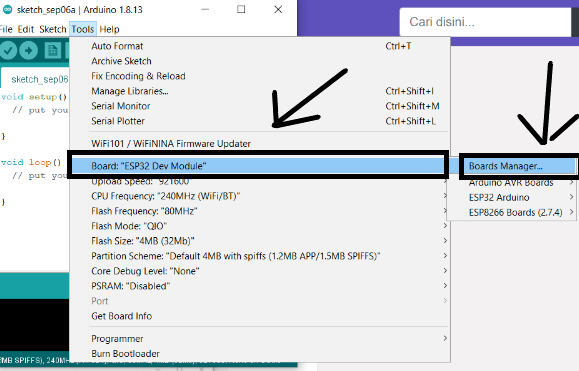
1. Selanjutnya buka File lalu klik Preferences



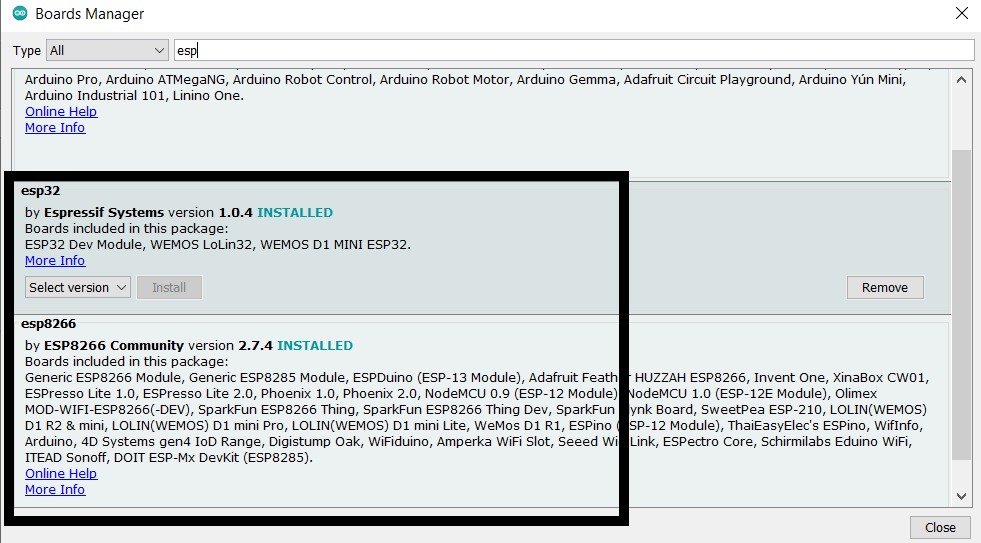
1. Lalu pada Additional Board Manager URLs: masukan link *http://arduino.esp8266.com/stable/package\_esp8266com\_index.json* , lalu klik Ok.



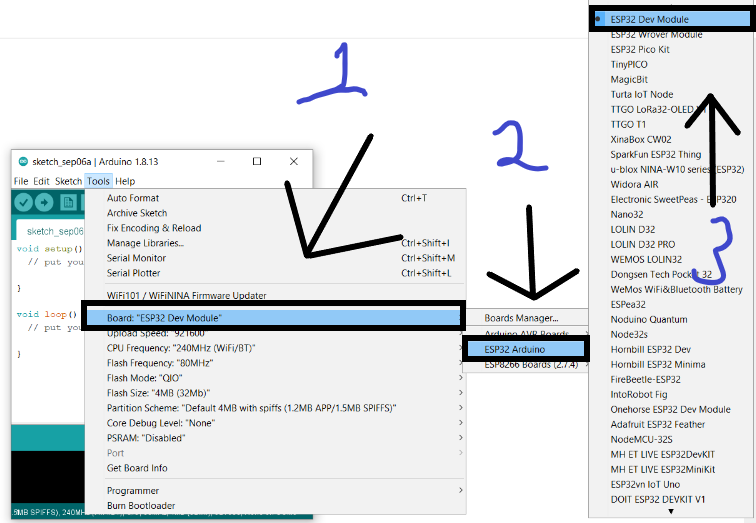
1. Lalu ke menu Tools, klik Board, lalu klik Boards Manager…



1. di Filter your search… atau dikolom pencarian masukan esp, klik esp8266 by ESP8266 Community, klik Install.



1. Tunggu sampai proses Install selesai, lalu tekan Close.
2. Setelah proses download selesai maka kita bisa menemukan board bernama nodemcu di submenu Board lalu pilih



1. Untuk memastikan bahwa kita berhasil menambahkan board nodemcu di arduino ide, kita bisa mencoba mengupload coding berikut ini ke nodemcu, pastikan upload speednya 115200 ya

Kode Program:

void setup() {

pinMode(LED\_BUILTIN, OUTPUT); // Initialize the LED\_BUILTIN pin as an output

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LED\_BUILTIN, LOW); // Turn the LED on (Note that LOW is the voltage level

|  |
| --- |
|  |

// but actually the LED is on; this is because

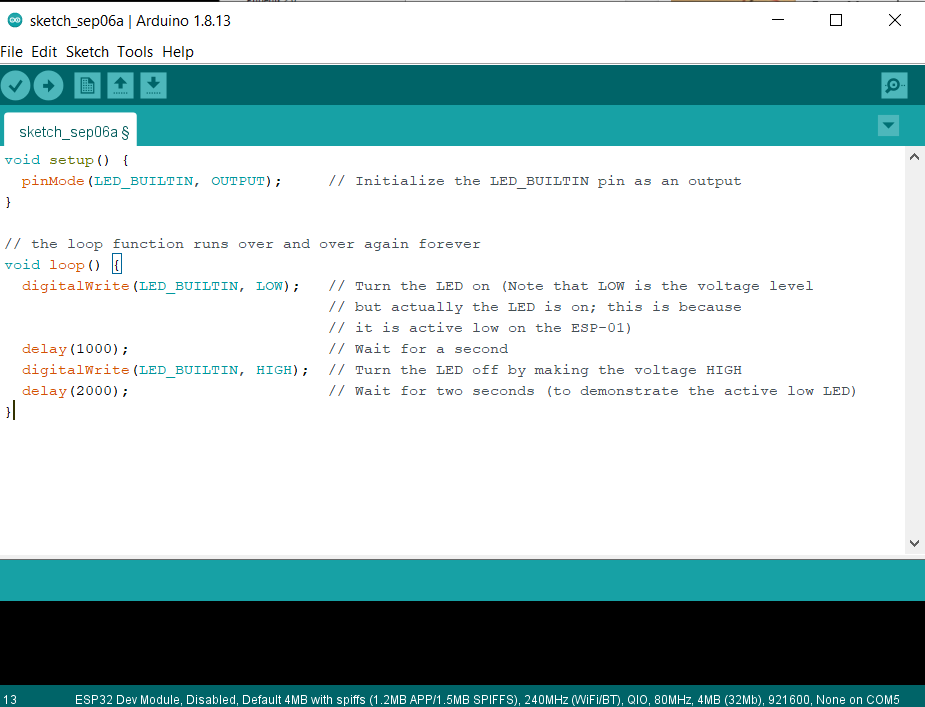
// it is active low on the ESP-01)

delay(1000); // Wait for a second

digitalWrite(LED\_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH

delay(2000); // Wait for two seconds (to demonstrate the active low LED)

}



1. jika berhasil diupload dan led pada board nodemcu berkedip maka selamat, berarti sekarang kita bisa memrogram board nodemcu pada Arduino IDE

**2.3.1.1 Kegunaan NodeMCU Pada Prototipe FiDo**

NodeMCU pada prototipe FiDo untuk pendeteksi kebakaran sebagai microcontroller yang mempunyai masukan dan keluaran. NodeMCU akan diprogram menggunakan Arduino IDE, kemudian setelah di upload pada nodemcu maka akan menghasilkan sebuah output untuk prototipe FiDo.

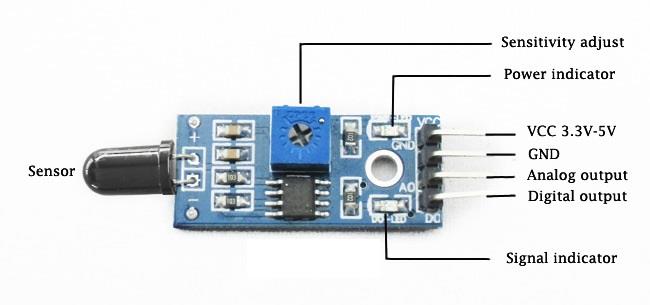
* + 1. **Sensor Api**

Sensor yang mempunyai fungsi untuk pendeteksi nyala api yang dimana api tersebut memiliki panjang gelombang antara 760nm – 1100nm. Sensor ini menggunakan infrared sebagai tranduser untuk kondisi nyala api.[15]

Skematik sensor flame detektor adalah rangkaian yang berfungsi mendeteksi keberadaan api pada jarak tertentu yang dapat di atur pada resistor variabel hingga jarak batas deteksi oleh fotodioda, apabilah sensor terdeteksi maka led merah pada sensor tersebut menyalah yang berarti sensor mendeteksi infrared yang di pancarkan oleh api.

Fitur- Fitur yang terdapat pada sensor api :

1. Tegangan operasi antara 3,3 – 5 Vdc
2. Terdapat 2 output yaitu digital output dan analog output yang berupa tegangan
3. Sudah terpackage dalam bentuk modul
4. Terdapat potensiometer sebagai pengaturan sensitivitas sensor dalam mensensing



*Gambar 2.4 Sensor Flame*

* + - 1. **Kegunaan Sensor Api pada Prototipe FiDo**

Sensor Api pada prototipe FiDo untuk pendeteksi kebakaran berfungsi sebagai pendeteksi nyala api. Dimana jarak maksimal sensor api dapat mendeteksi api sebesar 1 meter.

* + 1. **Sensor Asap**

Sensor MQ-2 atau sensor asap adalah sensor yang digunakan mendeteksi beberapa zat yaitu gas *LPG, i-butana, propana, metana, alkohol, hidrogen* dan asap. Inti daripada MQ-2 adalah material yang bersifat sensitif terhadap konsentrasi gas yang tersusun dari senyawa SnO2 atau disebut juga Timah (IV) Oksida. [16] Material ini mempunyai karakteristik akan berubah konduktivitasnya seiring dengan perubahan konsentrasi gas di sekitarnya**.** Jadi sensor asap ini sangat memudahkan dalam mendeteksi kejadian kebakaran.

Spesifikasi sensor Asap:

1. Tegangan kerja (Vcc): 5V
2. Lingkungan kerja:

suhu: 20℃±2℃

Kelembaban udara: 65%±5%

1. Range konsentrasi gas yang dapat diukur：

LPG dan propana: 200ppm-5000ppm

butana: 300ppm-5000ppm

metana: 5000ppm-20000ppm

Hidrogen: 300ppm-5000ppm

Alkohol: 100ppm-2000ppm



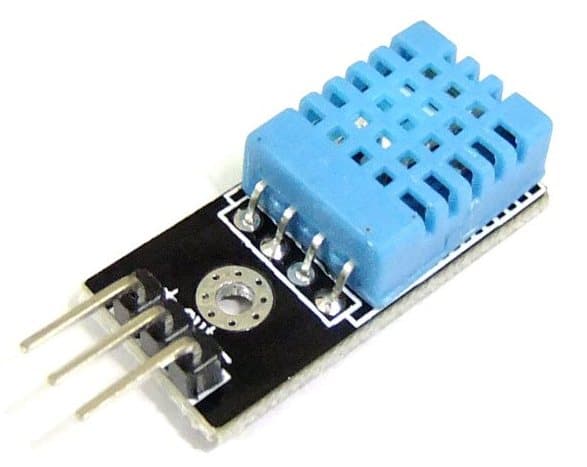
*Gambar 2.5 Sensor Asap*

* + - 1. **Kegunaan Sensor Asap pada Prototipe FiDo**

Sensor asap digunakan untuk mendeteksi asap yang dimana jika nilai adc (analog digital converter) lebih besar dari 800 maka akan terdeteksi asap sedang sedangkan jika dibawah 800 maka tidak ada asap dan jika diatas 1000 maka akan terdeteksi asap tinggi.

* + 1. **Sensor Suhu**

Sensor Suhu atau *Temperature Sensors* adalah suatu komponen yang dapat mengubah besaran panas menjadi besaran listrik sehingga dapat mendeteksi gejala perubahan suhu pada obyek tertentu. Sensor suhu melakukan pengukuran terhadap jumlah energi panas/dingin yang dihasilkan oleh suatu obyek sehingga memungkinkan kita untuk mengetahui atau mendeteksi gejala perubahan-perubahan suhu tersebut dalam bentuk output Analog maupun Digital. [16]



*Gambar 2.6 Sensor Suhu*

* + - 1. **Kegunaan Sensor Asap pada Prototipe FiDo**

Sensor suhu digunakan untuk mengetahui kondisi suhu yang ada disekitarnya yang dimana pada alat ini dikategorikan menjadi 3 yaitu suhu ruangan , suhu sedang dan suhu tinggi . Suhu ruangan berada dikisaran 25-30, suhu sedang berada di kisaran 31-35 dan suhu tinggi berada di 35 keatas.

* + 1. **Pompa Air**



*Gambar 2.6 Pompa Air*

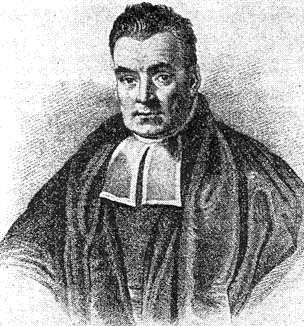
Pompa Air merupakan sebuah alat yang digunakan untuk Memindahkan cairan / air dari satu tempat ke tempat lain yang biasanya menggunakan energi listrik sebagai tenaga untuk mendorng air dengan cara menaikkan tekanan cairan tersebut untuk mengatasi hambatan pengaliran. Prinsip kerja pompa air mengubah energi motoric menjadi energi aliran Fluida [17]

Spesifikasi pompa air yang digunakan:

1. Tegangan kerja : 3~5V DC
2. Limit tegangan : 2.5 ~ 6V DC
3. Konsumsi Arus : 120-330mA
4. Konsumsi Daya : 0.4-1.5W
5. Kapasitas pompa : 80~120L/
6. Outside outlet: 7.5mm / 0.3"
7. Inside outlet: 4.7mm / 0.18"
8. Diameter Pompa : approx. 24mm / 0.95"
9. Panjang Pompa : approx. 45mm / 1.8"
10. Tinggi Pompa : approx. 33mm / 1.30"
11. Panjang kabel: 15-20cm
    * + 1. **Kegunaan Pompa Air pada Prototipe FiDo**

Pompa Air pada prototipe FiDo untuk mendeteksi kebakaran adalah untuk mengalirkan air ketika terjadi kebakaran atau nyala api sebagai mekanisme pemadaman kebakaran. Pompa air ini dapat mengalirkan air dalam waktu 3 detik sebanyak 200 ml.

* 1. **Metode Yang Digunakan** 
     1. **Metode Naïve Bayes**



Salah satu metode untuk mengklasifikasi dengan menggunakan teknik prediksi peluang kejadian yang sederhana dan mendasar. yang dimana dikemukakan oleh seorang ilmuwan Inggris yang bernama Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes. [18] Naive bayes untuk pada setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan dinyatakan benar, mengingat vektor informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat dalam menghasilkan perkiraan akhir dihitung sebagai jumlah frekuensi dari "master" tabel keputusan.

Berikut Langkah-langkah klasifikasi Metode Naïve Bayes

1. Menentukan data trainingnya
2. Menghitung probabilitas
3. Mencari Prediction Class
4. Mencari Confusion Class
5. Menghitung Akurasinya
   * + 1. **Prinsip Kerja Naïve Bayes**

Menurut Han dan Kamber Proses dari *The Naïve Bayesian classifier,* atau Simple Bayesian Classifier, sebagaiberikut:

1. Variable D akan menjadi pelatihan set tuple dan label yang terkait dengan kelas. Seperti biasa, setiap tuple diwakili oleh vector atribut n-dimensi, X = (x1, x2, ..., xn), ini menggambarkan pengukuran n dibuat pada tuple dari atribut n, masing-masing, A1, A2, ..., An.
2. Misalkan ada kelas m, C1, C2, ..., Cm. Akan diberi sebuah tuple, X,classifier akan memprediksi X yang masuk sebagai kelompok yang memiliki probabilitas posterior tertinggi, kondisi-disebutkan pada X. Artinya, classifier naive bayesian memprediksi bahwa X tuple milik kelas Ci jika dan hanya jika :

***P*(Ci|X) > *P*(Cj|X) for 1≤ j ≤ m, *j ≠ i***

Jadi memaksimalkan P (Ci| X). Ci kelas yang P (Ci | X) dimaksimalkan disebut sebagai hipotesis posteriori maksimal.

Dengan teorema Bayes:

***P*(Ci|X) =**

Ket:

P (Ci |X) = Probabilitas hipotesis Ci jika fakta atau record X (*Posterior Probability*)

P(X|Ci) = mencari nilai dari parameter yang memberi kemungkinan yang paling besar

P(Ci) = *Prior probability* dari pada X (*Prior probability*)

P(X) = Jumlah probability tuple yang akan muncul

1. Ketika P (X) adalah konstan pada semua kelas, hanya P (X | Ci) P (Ci) butuh dimaksimalkan. Jika probabilitas kelas sebelumnya tidak diketahui, maka umumnya diasumsikan ke dalam kelas yang sama, yaitu, P (C1) = P(C2) = · · · = P (Cm), maka dari itu akan memaksimalkan P (X | Ci). Jika tidak, maka akan memaksimalkan dengan P (X | Ci) P (Ci). Perhatikan bahwa probabilitas sebelum kelas dapat diperkirakan dengan P (Ci) = | Ci, D | / | D |, dimana | Ci, D | adalah jumlah tuple pelatihan kelas Ci di D.
2. Mengingat dataset mempunyai banyak atribut, sehingga akan sangat sulit dalam mengkomputasi untuk menghitung P(X|Ci). Agar dapat mengurangi perhitungan dalam mengevaluasi P(X|Ci), asumsi naïve independensi kelas bersyarat dibuat. Dianggap bahwa nilai-nilai dari atribut adalah kondisional independen satu sama lain, diberikan 15 kelas label dari tuple (yaitu bahwa tidak ada hubungan ketergantungan diantara atribut ) dengan demikian :

***P*(X|Ci) =**

**= P(x1|Ci) × P(x2|Ci) ×…× P(xn|Ci)**

Maka dapat dengan mudah diperkirakan probabilitas dengan P (x1 | Ci), P (x2 | Ci),. .. , P (xn | Ci) dari pelatihan tuple. Diingat Kembali bahwa di sini xk mengacu pada nilai atribut Ak untuk tuple X. Untuk setiap atribut, dilihat dari apakah atribut tersebut kategorikal atau continuous-valued. Misalnya, dalam menghitung P (X | Ci) mempertimbangkan hal-hal berikut:

1. Jika Ak merupakan kategorikal, maka P (Xk | Ci) adalah jumlah tuple kelas Ci di D memiliki nilai Xk untuk atribut Ak, dibagi dengan | Ci, D |, jumlah tuple kelas Ci di D.
2. Jika Ak *continuous-valued*, maka perlu melakukan lebih banyak pekerjaan, tapi perhitunganya lebih sederhana. Sebuah atribut *continuous-valued* biasanya diasumsikan memiliki distribusi Gaussian dengan rata-rata μ dan standar deviasi σ, didefinisikan oleh

Sehingga

***P* (xk| Ci) *=g (xk, µci, )***

Setelah itu hitung μCi dan σCi , yang merupakan deviasi mean (rata-rata) dan standar masing-masing nilai atribut Ak untuk tuple pelatihan kelas Ci . Setelah itu menggunakan kedua kuantitas dalam Persamaan, bersama-sama dengan xk, untuk memperkirakan P (xk | Ci)

1. Untuk memprediksi label kelas x, P(X|Ci) P(Ci) dievaluasi untuk setiap kelas Ci.Classifier memprediksi kelas label dari tuple x adalah kelas Ci, jika

Dengan kata lain, label kelas dapat diprediksi adalah Ci yang mana P (X | Ci) P (Ci) adalah maksimal.

* + - 1. **Kelebihan Dan Kekurangan Naïve Bayes**

**Kelebihan:**

1. Mudah untuk dipahami
2. Hanya memerlukan pengkodean yang sederhana
3. Lebih cepat dalam penghitungan
4. Menangani kuantitatif dan data diskrit
5. Kokoh untuk titik noise yang diisolasi, misalkan titik yang dirata – ratakan ketika mengestimasi peluang bersyarat data.
6. Hanya membutuhkan sejumlah kecil data pelatihan (data *training*) untuk mengestimasi parameter yang dibutuhkan untuk klasifikasi.

**Kekurangan:**

1. Tidak akan terjadi jika probabilitas kondisionalnya adalah nol, apabila nol maka probabilitas prediksi akan bernilai nol juga
2. Mengasumsikan variabel bebas
   1. **Tinjauan Pustaka** 
      * 1. E. Junianto dan R. Rachman. "Penerapan Metode Naïve Bayes Classifier Untuk Mendeteksi Emosi Pada Komentar Media Sosial". Jurnal Responsif,Vol. 2, No. 1, Februari 2020 . Penelitian ini dilakukan untuk mengetahui seberapa baik NBC untuk mengklasifikasikan Emosi dari data yang berbentuk teks dengan menggunakan metode Naïve Bayes [19]. Perbedaan dengan penelitan ini adalah metode Naïve Bayes digunakan untuk Tetapi alat – alat tersebut tidak dapat menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.
        2. K. Anwar, D. Shaugy,dan H. Fitriyah, "Sistem Pendeteksi Kandungan Nutrisi dalam Tanah Berdasarkan Warna dan Kelembapan dengan Menggunakan Metode Naive Bayes," .Jurnal Pengembangan Teknologi Informasi dan Ilmu Kompute*r,* Vol. 2, No. 9, September 2018. Metode Naive Bayes cukup efektif dalam melakukan proses klasifikasi kandungan nutrisi tanah, terbukti dengan hasil pengujian akurasi sistem yang dapat melakukan klasifikasi berdasarkan kelas yang telah ditentukan oleh pakar, yaitu kandungan nutrisi rendah dan kandungan nutrisi tinggi[20]. Perbedaan dengan penelitan ini adalah metode Naïve Bayes digunakan untuk mengklasifikasikan kodisi atau status dari suatu ruangan dalam kondisi aman atau bahaya
        3. Mohamad Misfaul May Dana, Wijaya Kurniawan, dan Hurryatul Fitriya, "Rancang Bangun Sistem Deteksi Titik Kebakaran Dengan Metode Naive Menggunakan Sensor Asap dan Sensor Api Berbasis Arduino," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*,* Vol. 2, No. 9, September 2018. Pada penelitian ini telah dibuat sistem otomatisasi untuk mendeteksi lokasi titik kebakaran berdasarkan nilai Asap ruangan dengan menggunakan metode Naíve Bayes.Dimana baik semua komponen alat yang digunakan maupun metode Naíve Bayes yang diterapkan dapat berjalan sesuai dengan yang diinginkan, terbukti dengan sistem dapat mengklasifikasikan lokasi titik kebakaran yaitu, daerah 1, daerah 2, daerah 3, dan daerah 4. [21] Perbedaan dengan penelitan ini adalah metode Naïve Bayes digunakan untuk dapat menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.
        4. Dwi Susanto, Achmad Basuki dan Prada Duanda, “Deteksi Plagiat Dokumen Tugas Daring Laporan Praktikum Mata Kuliah Desain Web Menggunakan Metode Naive Bayes” Nusantara Journal of Computers and its Applications, Vol. 2, No. 1 Desember 2016.Dalam penelitan ini Naïve Bayes digunakan untuk mendeteksi kesamaan antar dokumen tugas mahasiswa. Hasil yang didapatkan dari sistem pendeteksi plagiat ini adalah persentase kesamaan antara dokumen yang dibandingkan. [22] Perbedaan dengan penelitan ini adalah metode Naïve Bayes digunakan untuk menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.
        5. Rint Zata, Amani1 Rizal Maulana, dan Dahnial Syauqy, “Sistem Pendeteksi Dehidrasi Berdasarkan Warna dan Kadar Amonia pada Urin Berbasis Sensor TCS3200 Dan MQ135 dengan Metode Naive Bayes” Vol. 1, No. 5, Mei 2017. Metode Naive Bayes yang diterapkan dapat berjalan sesuai dengan yang diinginkan, terbukti dengan sistem dapat mengklasifikasikan jenis dehidrasi menjadi dehidrasi ringan, dehidrasi sedang dan dehidrasi berat [23]. Perbedaan dengan penelitan ini adalah metode Naïve Bayes digunakan untuk menentukan kondisi suatu ruangan dalam kondisi aman atau bahaya.