



[Main page](#)
[Recent changes](#)
[Random page](#)
[Help about MediaWiki](#)

Tools

[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)
[Page information](#)

Page

[Discussion](#)

Read

[View source](#)

More ▾

Search



Domain Name System

DNS (*Domain Name System*, **bahasa Indonesia**: **Sistem Penamaan Domain**) adalah sebuah sistem yang menyimpan informasi tentang **nama host** maupun **nama domain** dalam bentuk **basis data tersebar** (*distributed database*) di dalam jaringan komputer, misalkan: **Internet**. DNS menyediakan **alamat IP** untuk setiap nama host dan mendata setiap **server transmisi surat** (mail exchange server) yang menerima surat elektronik (*email*) untuk setiap domain.

DNS menyediakan servis yang cukup penting untuk Internet, bilamana **perangkat keras komputer** dan jaringan bekerja dengan **alamat IP** untuk mengerjakan tugas seperti pengalamatan dan penjaluran (**routing**), manusia pada umumnya lebih memilih untuk menggunakan nama host dan nama domain, contohnya adalah penunjukan sumber universal (**URL**) dan **alamat e-mail**. **DNS** menghubungkan kebutuhan ini.

Contents [hide]

- 1 Sejarah singkat DNS
- 2 Teori Kerja DNS
 - 2.1 Para Pemain Inti
 - 2.2 Pengertian beberapa bagian dari nama domain
 - 2.3 Sebuah contoh dari teori rekursif DNS
 - 2.4 Pengertian pendaftaran domain dan *glue records*
- 3 DNS dalam praktek
 - 3.1 Caching dan masa hidup (*caching and time to live*)
 - 3.2 Waktu propagasi (*propagation time*)
 - 3.3 DNS di dunia nyata
 - 3.4 Penerapan DNS lainnya
- 4 Jenis-jenis catatan DNS
- 5 Nama domain yang diinternasionalkan
- 6 Perangkat lunak DNS
- 7 Pengguna legal dari domain
 - 7.1 Pendaftar (*registrant*)
 - 7.2 Kontak Administratif (*Administrative Contact*)
 - 7.3 Kontak Teknis (*Technical Contact*)
 - 7.4 Kontak Pembayaran (*Billing Contact*)
 - 7.5 Server Nama (*Name Servers*)
- 8 Politik
- 9 Lihat pula
- 10 YOUTUBE
- 11 Pranala luar dan dokumentasi
- 12 Referensi

Sejarah singkat DNS

Penggunaan nama sebagai pengabstraksi alamat mesin di sebuah [jaringan komputer](#) yang lebih dikenal oleh manusia mengalahkan [TCP/IP](#), dan kembali ke zaman [ARPAnet](#). Dahulu, setiap komputer di jaringan komputer menggunakan file **HOSTS.TXT** dari SRI (sekarang [SRI International](#)), yang memetakan sebuah alamat ke sebuah nama (secara teknis, file ini masih ada - sebagian besar sistem operasi modern menggunakannya baik secara baku maupun melalui konfigurasi, dapat melihat [Hosts file](#) untuk menyamakan sebuah [nama host](#) menjadi sebuah [alamat IP](#) sebelum melakukan pencarian via [DNS](#)). Namun, sistem tersebut diatas mewarisi beberapa keterbatasan yang mencolok dari sisi prasyarat, setiap saat sebuah alamat [komputer](#) berubah, setiap sistem yang hendak berhubungan dengan komputer tersebut harus melakukan update terhadap file Hosts.

Dengan berkembangnya [jaringan komputer](#), membutuhkan sistem yang bisa dikembangkan: sebuah sistem yang bisa mengganti alamat host hanya di satu tempat, host lain akan mempelajari perubahan tersebut secara dinamis. Inilah [DNS](#).

[Paul Mockapetris](#) menemukan [DNS](#) di tahun [1983](#); spesifikasi asli muncul di [RFC 882](#) dan [883](#). Tahun [1987](#), penerbitan [RFC 1034](#) dan [RFC 1035](#) membuat update terhadap spesifikasi [DNS](#). Hal ini membuat [RFC 882](#) dan [RFC 883](#) tidak berlaku lagi. Beberapa RFC terkini telah memproposisikan beberapa tambahan dari protokol inti [DNS](#).

Teori Kerja DNS

Para Pemain Inti

Pengelola dari sistem DNS terdiri dari tiga komponen:

- **DNS resolver**, sebuah program klien yang berjalan di komputer pengguna, yang membuat permintaan DNS dari program aplikasi.
- **recursive DNS server**, yang melakukan pencarian melalui DNS sebagai tanggapan permintaan dari *resolver*, dan mengembalikan jawaban kepada para *resolver* tersebut;

dan ...

- **authoritative DNS server** yang memberikan jawaban terhadap permintaan dari *resolver*, baik dalam bentuk sebuah jawaban, maupun dalam bentuk delegasi (misalkan: mereferensikan ke *authoritative DNS server* lainnya)

Pengertian beberapa bagian dari nama domain

Sebuah [nama domain](#) biasanya terdiri dari dua bagian atau lebih (secara teknis disebut *label*), dipisahkan dengan titik.

- Label paling kanan menyatakan **top-level domain** - domain tingkat atas/tinggi (misalkan, alamat [www.wikipedia.org](#) memiliki top-level domain [org](#)).

- Setiap label di sebelah kirinya menyatakan sebuah sub-divisi atau **subdomain** dari domain yang lebih tinggi. Catatan: "subdomain" menyatakan ketergantungan relatif, bukan absolut. Contoh: wikipedia.org merupakan subdomain dari domain org, dan id.wikipedia.org dapat membentuk subdomain dari domain wikipedia.org (pada prakteknya, id.wikipedia.org sesungguhnya mewakili sebuah nama host - lihat dibawah). Secara teori, pembagian seperti ini dapat mencapai kedalaman 127 level, dan setiap label dapat terbentuk sampai dengan 63 karakter, selama total nama domain tidak melebihi panjang 255 karakter. Tetapi secara praktek, beberapa **pendaftar nama domain** (*domain name registry*) memiliki batas yang lebih sedikit.
- Terakhir, bagian paling kiri dari bagian nama domain (biasanya) menyatakan nama host. Sisa dari nama domain menyatakan cara untuk membangun jalur logis untuk informasi yang dibutuhkan; nama host adalah tujuan sebenarnya dari nama sistem yang dicari alamat IP-nya. Contoh: nama domain www.wikipedia.org memiliki nama host "www".

DNS memiliki kumpulan hirarki dari **DNS servers**. Setiap domain atau subdomain memiliki satu atau lebih **authoritative DNS Servers** (server DNS otoritatif) yang mempublikasikan informasi tentang domain tersebut dan nama-nama **server** dari setiap domain di-"bawah"-nya. Pada puncak hirarki, terdapat **root servers**- induk server nama: server yang ditanyakan ketika mencari (**menyelesaikan/resolving**) dari sebuah nama domain tertinggi (*top-level domain*).

Sebuah contoh dari teori rekursif DNS

Sebuah contoh mungkin dapat memperjelas proses ini. Andaikan ada aplikasi yang memerlukan pencarian alamat IP dari www.wikipedia.org. Aplikasi tersebut bertanya ke *DNS recursor* lokal.

- Sebelum dimulai, *recursor* harus mengetahui dimana dapat menemukan *root nameserver*; administrator dari *recursive DNS server* secara manual mengatur (dan melakukan update secara berkala) sebuah file dengan nama **root hints zone** (panduan akar DNS) yang menyatakan alamat-alamat IP dari para server tersebut.
- Proses dimulai oleh *recursor* yang bertanya kepada para *root server* tersebut - misalkan: server dengan alamat IP "198.41.0.4" - pertanyaan "apakah alamat IP dari www.wikipedia.org?"
- *Root server* menjawab dengan sebuah **delegasi**, arti kasarnya: "Saya tidak tahu alamat IP dari www.wikipedia.org, tapi saya "tahu" bahwa server DNS di 204.74.112.1 memiliki informasi tentang domain org."
- *Recursor* DNS lokal kemudian bertanya kepada server DNS (yaitu: 204.74.112.1) pertanyaan yang sama seperti yang diberikan kepada *root server*. "apa alamat IP dari www.wikipedia.org?". (umumnya) akan didapatkan jawaban yang sejenis, "saya tidak tahu alamat dari www.wikipedia.org, tapi saya "tahu" bahwa server 207.142.131.234 memiliki informasi dari domain wikipedia.org."
- Akhirnya, pertanyaan beralih kepada server DNS ketiga (207.142.131.234),

yang menjawab dengan alamat IP yang dibutuhkan.

Proses ini menggunakan **pencarian rekursif** (*recursion / recursive searching*).

Pengertian pendaftaran domain dan *glue records*

Membaca contoh diatas, Anda mungkin bertanya: "bagaimana caranya DNS server 204.74.112.1 tahu alamat IP mana yang diberikan untuk domain wikipedia.org?" Pada awal proses, kita mencatat bahwa sebuah *DNS recursor* memiliki alamat IP dari para *root server* yang (kurang-lebih) didata secara eksplisit (*hard coded*). Mirip dengan hal tersebut, server nama (*name server*) yang otoritatif untuk *top-level domain* mengalami perubahan yang jarang.

Namun, server nama yang memberikan jawaban otoritatif bagi nama domain yang umum mengalami perubahan yang cukup sering. Sebagai bagian dari proses pendaftaran sebuah nama domain (dan beberapa waktu sesudahnya), pendaftar memberikan pendaftaran dengan server nama yang akan mengotorisasikan nama domain tersebut; maka ketika mendaftar wikipedia.org, domain tersebut terhubung dengan server nama gunther.bomis.com dan zwinger.wikipedia.org di pendaftar .org. Kemudian, dari contoh di atas, ketika server dikenali sebagai 204.74.112.1 menerima sebuah permintaan, DNS server memindai daftar domain yang ada, mencari wikipedia.org, dan mengembalikan server nama yang terhubung dengan domain tersebut.

Biasanya, server nama muncul berdasarkan urutan nama, selain berdasarkan alamat IP. Hal ini menimbulkan *string* lain dari permintaan DNS untuk menyelesaikan nama dari server nama; ketika sebuah alamat IP dari server nama mendapatkan sebuah pendaftaran di zona induk, para programmer jaringan komputer menamakannya sebuah **glue record** (daftar lekat???)

DNS dalam praktek

Ketika sebuah aplikasi (misalkan web browser), hendak mencari alamat IP dari sebuah nama domain, aplikasi tersebut tidak harus mengikuti seluruh langkah yang disebutkan dalam *teori* diatas. Kita akan melihat dulu konsep *caching*, lalu mengertikan operasi DNS di "dunia nyata".

Caching dan masa hidup (*caching and time to live*)

Karena jumlah permintaan yang besar dari sistem seperti DNS, perancang DNS menginginkan penyediaan mekanisme yang bisa mengurangi beban dari masing-masing server DNS. Rencana mekanisnya menyarankan bahwa ketika sebuah *DNS resolver* (klien) menerima sebuah jawaban DNS, informasi tersebut akan di **cache** untuk jangka waktu tertentu. Sebuah nilai (yang di-set oleh administrator dari server DNS yang memberikan jawaban) menyebutnya sebagai **time to live** (masa hidup), atau **TTL** yang mendefinisikan periode tersebut. Saat jawaban masuk ke dalam *cache*, *resolver* akan mengacu kepada jawaban yang disimpan di *cache* tersebut; hanya ketika TTL usai (atau saat administrator mengosongkan jawaban dari memori *resolver* secara manual) maka *resolver* menghubungi server DNS untuk informasi yang sama.

Waktu propagasi (*propagation time*)

Satu akibat penting dari arsitektur tersebar dan *cache* adalah perubahan kepada suatu DNS tidak selalu efektif secara langsung dalam skala besar/global. Contoh berikut mungkin akan menjelaskannya: Jika seorang administrator telah mengatur **TTL** selama 6 jam untuk host www.wikipedia.org, kemudian mengganti alamat IP dari www.wikipedia.org pada pk 12:01, administrator harus mempertimbangkan bahwa ada (paling tidak) satu individu yang menyimpan *cache* jawaban dengan nilai lama pada pk 12:00 yang tidak akan menghubungi server DNS sampai dengan pk 18:00. Periode antara pk 12:00 dan pk 18:00 dalam contoh ini disebut sebagai **waktu propagasi** (*propagation time*), yang bisa didefinisikan sebagai periode waktu yang berawal antara saat terjadi perubahan dari data DNS, dan berakhir sesudah waktu maksimum yang telah ditentukan oleh **TTL** berlalu. Ini akan mengarahkan kepada pertimbangan logis yang penting ketika membuat perubahan kepada DNS: *tidak semua akan melihat hal yang sama seperti yang Anda lihat*. [RFC1537](#) dapat membantu penjelasan ini.

DNS di dunia nyata

Di dunia nyata, user tidak berhadapan langsung dengan *DNS resolver* - mereka berhadapan dengan program seperti **web browser** ([Mozilla Firefox](#), [Safari](#), [Opera](#), [Internet Explorer](#), [Netscape](#), [Konqueror](#) dan lain-lain dan klien mail ([Outlook Express](#), [Mozilla Thunderbird](#) dan lain-lain). Ketika user melakukan aktivitas yang meminta pencarian DNS (umumnya, nyaris semua aktivitas yang menggunakan Internet), program tersebut mengirimkan permintaan ke *DNS Resolver* yang ada di dalam [sistem operasi](#).

DNS resolver akan selalu memiliki *cache* (lihat diatas) yang memiliki isi pencarian terakhir. Jika *cache* dapat memberikan jawaban kepada permintaan DNS, *resolver* akan menggunakan nilai yang ada di dalam *cache* kepada program yang memerlukan. Kalau *cache* tidak memiliki jawabannya, *resolver* akan mengirimkan permintaan ke server DNS tertentu. Untuk kebanyakan pengguna di rumah, [Internet Service Provider\(ISP\)](#) yang menghubungkan komputer tersebut biasanya akan menyediakan server DNS: pengguna tersebut akan mendata alamat server secara manual atau menggunakan [DHCP](#) untuk melakukan pendataan tersebut. Jika administrator sistem telah mengkonfigurasi sistem untuk menggunakan server DNS mereka sendiri, *DNS resolver* umumnya akan mengacu ke server nama mereka. Server nama ini akan mengikuti proses yang disebutkan di *Teori DNS*, baik mereka menemukan jawabannya maupun tidak. Hasil pencarian akan diberikan kepada *DNS resolver*; diasumsikan telah ditemukan jawaban, *resolver* akan menyimpan hasilnya di *cache* untuk penggunaan berikutnya, dan memberikan hasilnya kepada software yang meminta pencarian DNS tersebut.

Sebagai bagian akhir dari kerumitan ini, beberapa aplikasi seperti *web browser* juga memiliki *DNS cache* mereka sendiri, tujuannya adalah untuk mengurangi penggunaan referensi *DNS resolver*, yang akan meningkatkan kesulitan untuk melakukan **debug** DNS, yang menimbulkan kerancuan data yang lebih akurat. *Cache* seperti ini umumnya memiliki masa yang singkat dalam hitungan 1 menit.

Penerapan DNS lainnya

Sistem yang dijabarkan diatas memberikan skenario yang disederhanakan. DNS meliputi beberapa fungsi lainnya:

- Nama host dan alamat IP tidak berarti terhubung secara satu-banding-satu. Banyak nama host yang diwakili melalui alamat IP tunggal: gabungan dengan **pengasuhan maya** (*virtual hosting*), hal ini memungkinkan satu komputer untuk melayani beberapa situs web. Selain itu, sebuah nama host dapat mewakili beberapa alamat IP: ini akan membantu toleransi kesalahan (**fault tolerance** dan penyebaran beban (*load distribution*), juga membantu suatu situs berpindah dari satu lokasi fisik ke lokasi fisik lainnya secara mudah.
- Ada cukup banyak kegunaan DNS selain menerjemahkan nama ke alamat IP. Contoh:, agen pemindahan surat **Mail transfer agents(MTA)** menggunakan DNS untuk mencari tujuan pengiriman **E-mail** untuk alamat tertentu. Domain yang menginformasikan pemetaan *exchange* disediakan melalui rekod MX (**MX record**) yang meningkatkan lapisan tambahan untuk toleransi kesalahan dan penyebaran beban selain dari fungsi pemetaan nama ke alamat IP.
- Kerangka Peraturan Pengiriman (**Sender Policy Framework**) secara kontroversi menggunakan keuntungan jenis rekod DNS, dikenal sebagai rekod TXT.
- Menyediakan keluwesan untuk kegagalan komputer, beberapa server DNS memberikan perlindungan untuk setiap domain. Tepatnya, tigabelas server akar (*root servers*) digunakan oleh seluruh dunia. Program DNS maupun sistem operasi memiliki alamat IP dari seluruh server ini. Amerika Serikat memiliki, secara angka, semua kecuali tiga dari server akar tersebut. Namun, dikarenakan banyak server akar menerapkan **anycast**, yang memungkinkan beberapa komputer yang berbeda dapat berbagi alamat IP yang sama untuk mengirimkan satu jenis *services* melalui area geografis yang luas, banyak server yang secara fisik (bukan sekedar angka) terletak di luar Amerika Serikat.

DNS menggunakan **TCP** dan **UDP** di **port komputer** 53 untuk melayani permintaan DNS. Nyaris semua permintaan DNS berisi permintaan UDP tunggal dari klien yang diikuti oleh jawaban UDP tunggal dari server. Umumnya TCP ikut terlibat hanya ketika ukuran data jawaban melebihi 512 byte, atau untuk pertukaran zona DNS **zone transfer**

Jenis-jenis catatan DNS

Beberapa kelompok penting dari data yang disimpan di dalam DNS adalah sebagai berikut:

- **A record** atau **catatan alamat** memetakan sebuah nama host ke alamat IP 32-bit (untuk **IPv4**).
- **AAAA record** atau **catatan alamat IPv6** memetakan sebuah nama host ke alamat IP 128-bit (untuk **IPv6**).
- **CNAME record** atau **catatan nama kanonik** membuat alias untuk nama domain. Domain yang di-alias-kan memiliki seluruh subdomain dan rekod DNS seperti aslinya.

- **MX record** atau **catatan pertukaran surat** memberikan informasi tentang *mail exchange server* / mail server untuk sebuah domain.
- **PTR record** atau **catatan penunjuk** memetakan sebuah nama host ke nama kanonik untuk host tersebut. Pembuatan rekod PTR untuk sebuah nama host di dalam domain `in-addr.arpa` yang mewakili sebuah alamat IP menerapkan pencarian balik DNS (*reverse DNS lookup*) untuk alamat tersebut. Contohnya (saat penulisan / penerjemahan artikel ini), `www.icann.net` memiliki alamat IP `192.0.34.164`, tetapi sebuah rekod PTR memetakan `164.34.0.192.in-addr.arpa` ke nama kanoniknya: `referrals.icann.org`.
- **NS record** atau **catatan server nama** memberitahukan daftar name server (NS) yang membawa informasi tentang sebuah domain.
- **SOA record** atau **catatan otoritas awal** (*Start of Authority*) memberikan informasi server DNS yang mempunyai otoritas tertinggi untuk sebuah domain.
- **SRV record** adalah catatan lokasi secara umum.
- Catatan **TXT** mengizinkan administrator untuk memasukan data acak ke dalam catatan DNS; catatan ini juga digunakan di spesifikasi *Sender Policy Framework*.

Jenis catatan lainnya semata-mata untuk penyediaan informasi (contohnya, catatan **LOC** memberikan letak *lokasi* fisik dari sebuah host, atau data uji coba (misalkan, catatan **WKS** memberikan sebuah daftar dari server yang memberikan servis yang dikenal (*well-known service*) seperti HTTP atau POP3 untuk sebuah domain.

Nama domain yang diinternasionalkan

Nama domain harus menggunakan satu sub-kumpulan dari karakter **ASCII**, hal ini mencegah beberapa bahasa untuk menggunakan nama maupun kata lokal mereka. **ICANN** telah menyetujui **Punycode** yang berbasiskan sistem **IDNA**, yang memetakan string **Unicode** ke karakter set yang valid untuk DNS, sebagai bentuk penyelesaian untuk masalah ini, dan beberapa **registries** sudah mengadopsi metode IDNS ini.

Perangkat lunak DNS

Beberapa jenis **perangkat lunak DNS** menerapkan metode **DNS**, beberapa diantaranya:

- **BIND** (Berkeley Internet Name Domain)
- **djbdns** (**Daniel J. Bernstein's** DNS)
- **MaraDNS**
- **QIP** (Lucent Technologies)
- **NSD** (Name Server Daemon)
- **PowerDNS**

Utiliti berorientasi **DNS** termasuk:

- dig (the domain information groper)

Pengguna legal dari domain

Pendaftar (*registrant*)

Tidak satupun individu di dunia yang "memiliki" nama domain kecuali **Network Information Centre** (NIC), atau pendaftar nama domain (*domain name registry*). Sebagian besar dari NIC di dunia menerima biaya tahunan dari para pengguna legal dengan tujuan bagi si pengguna legal menggunakan nama domain tersebut. Jadi sejenis perjanjian sewa-menyewa terjadi, bergantung kepada syarat dan ketentuan pendaftar. Bergantung kepada beberapa peraturan penamaan dari para pendaftar, pengguna legal dikenal sebagai "pendaftar" (*registrants*) atau sebagai "pemegang domain" (*domain holders*)

ICANN memegang daftar lengkap untuk pendaftar domain di seluruh dunia. Siapapun dapat menemukan pengguna legal dari sebuah domain dengan mencari melalui basis data **WHOIS** yang disimpan oleh beberapa pendaftar domain.

Di (lebih kurang) 240 **country code top-level domains** (ccTLDs), pendaftar domain memegang sebuah acuan WHOIS (pendaftar dan nama server). Contohnya, **IDNIC**, NIC Indonesia, memegang informasi otoritatif WHOIS untuk nama domain .ID.

Namun, beberapa pendaftar domain, seperti **VeriSign**, menggunakan model pendaftar-pengguna. Untuk nama domain .COM dan .NET, pendaftar domain, VeriSign memegang informasi dasar WHOIS (pemegang domain dan server nama). Siapapun dapat mencari detil **WHOIS** (Pemegang domain, server nama, tanggal berlaku, dan lain sebagainya) melalui pendaftar.

Sejak sekitar 2001, kebanyakan pendaftar **gTLD** (.ORG, .BIZ, .INFO) telah mengadopsi metode penfatar "tebal", menyimpan otoritatif **WHOIS** di beberapa pendaftar dan bukan pendaftar itu saja.

Kontak Administratif (*Administrative Contact*)

Satu pemegang domain biasanya menunjuk kontak administratif untuk menangani nama domain. Fungsi manajemen didelegasikan ke kontak administratif yang mencakup (diantaranya):

- keharusan untuk mengikuti syarat dari pendaftar domain dengan tujuan memiliki hak untuk menggunakan nama domain
- otorisasi untuk melakukan update ke alamat fisik, alamat **email** dan nomor telepon dan lain sebagainya via **WHOIS**

Kontak Teknis (*Technical Contact*)

Satu kontak teknis menangani server nama dari sebuah nama domain. Beberapa dari banuak fungsi kontak teknis termasuk:

- memastikan bahwa konfigurasi dari nama domain mengikuti syarat dari pendaftar domain

- [update zona domain](#)
- menyediakan fungsi 24x7 untuk ke server nama (yang membuat nama domain bisa diakses)

Kontak Pembayaran (*Billing Contact*)

Tidak perlu dijelaskan, pihak ini adalah yang menerima tagihan dari [NIC](#).

Server Nama (*Name Servers*)

Disebut sebagai server nama otoritatif yang mengasuh zona nama domain dari sebuah nama domain.

Politik

Banyak penyelidikan telah menyuarakan kritik dari metode yang digunakan sekarang untuk mengatur kepemilikan domain. Umumnya, kritik mengklaim penyalahgunaan dengan monopoli, seperti [VeriSign](#) Inc dan masalah-masalah dengan penunjukkan dari [top-level domain](#) (TLD). Lembaga internasional [ICANN](#) (*Internet Corporation for Assigned Names and Numbers*) memelihara industri nama domain.

Lihat pula

- [cybersquatting](#)
- [dynamic DNS](#)
- [DNSSEC](#)
- [ICANN](#)
- [Root nameserver](#)
- [PANDI](#)
- [DNSCrypt](#)

YOUTUBE

- [YOUTUBE: Konsep Domain Name](#)[↗](#)

Pranala luar dan dokumentasi

- [Contoh DNS](#)[↗](#)
- [Pemalsuan DNS](#)[↗](#)
- [Pemalsuan DNS](#)[↗](#)
- [Situs dukungan DNS LOC](#)[↗](#)
- [Alat pencarian DNS dengan pengukur kecepatan akses](#)[↗](#)
- [Bagaimana melakukan pemblokiran iklan dengan DNS dan file hosts](#)[↗](#)
- [Signposts in Cyberspace: The Domain Name System and Internet Navigation](#) (PDF format)[↗](#)
- [Alat pencarian online sederhana seperti whois, reverse whois, dan rekod A serta MX](#)[↗](#)

- [DNS untuk Rocket Scientists oleh zytrax.com](#)
- [Domain Name System Links, Whitepapers, dan Research](#)

Referensi

- <http://www.domain.my.id/> - daftar registrar Indonesia

Pranala Menarik

- [DNS: Pengertian Sederhana](#)
- [Fully Qualified Domain Name](#)
- [PowerDNS](#)
- [Instalasi BIND](#)
- [BIND: Instalasi dan Konfigurasi](#)
- [BIND RPZ](#)
- [Implementasi ENUM Server](#)
- [tinydns](#)
- [Nawala Project](#)
- [Daftar DNS Server Bagi Pengguna Indonesia](#)
- [DNS: Spoofing ntp.ubuntu.com](#)
- [DNS: Spoofing debian.pool.ntp.org](#)
- [Webmin: Konfigurasi DNS](#)
- <http://myubuntutux.blogspot.com/2011/11/mempercepat-internet-ubuntu.html>
- <http://numb.web.id/ubuntu/cara-setting-dns-server-yang-baik-di-ubuntu-11-04.html>
- [DNS: Root Server](#)
- [DNS: Linux systemd-resolved](#)
- [BIND: PTR record](#)

This page was last modified on 15 September 2020, at 09:39.

[Privacy policy](#) [About OnnoWiki](#) [Disclaimers](#)

