

DSDN: Jaringan data dan komputasi semi desentral dan terdistribusi dengan blockchain



Tanggal: 16 Oktober 2025

Penulis: Noven rizkia

novenrizkia.8.5.6@gmail.com

Atas nama:Ineva Researcher

Abstrak: Kami mengusulkan DSDN, jaringan data dan komputasi semi-desentral yang menggabungkan replikasi data lintas zona, eksekusi program terisolasi, dan lapisan validator untuk kepatuhan hukum. DSDN menyediakan penempatan data berbasis content addressing, target replikasi tiga kali, eksekusi Rust/Python di atas WASM atau microVM, serta penjadwalan yang memperhitungkan lokasi data dan sumber daya. Validator memeriksa aplikasi yang melayani publik tanpa mengakses data privat yang terenkripsi. Tujuan utama adalah ketersediaan tinggi, biaya yang terukur, dan partisipasi masyarakat luas, dengan ruang bagi pemerintah sebagai pengawas konten ilegal, tidak sebagai pengendali data.

DSDN dirancang sebagai sistem *verifiable-by-design*, di mana tidak ada satu pun entitas, termasuk coordinator, validator, maupun foundation, yang memegang kontrol otoritatif atas data, eksekusi, maupun state jaringan. Seluruh metadata sistem direkam pada Data Availability layer yang bersifat publik dan dapat direplay secara deterministik, sehingga setiap node, auditor independen, atau pihak ketiga dapat memverifikasi kebenaran state jaringan

tanpa memerlukan kepercayaan implisit kepada operator mana pun. Dengan pendekatan ini, DSDN memprioritaskan auditabilitas, transparansi, dan pembatasan kekuasaan struktural sebagai fondasi kepercayaan jangka panjang.

1. Pendahuluan

Internet modern berjalan di atas infrastruktur yang sangat terkonsentrasi. Sebagian besar data, komputasi, dan layanan digital dunia dioperasikan oleh sejumlah kecil pusat data global. Model ini memberikan efisiensi skala, tetapi juga menciptakan konsentrasi kekuasaan struktural, risiko sensor, ketergantungan geopolitik, serta single point of failure yang bersifat sistemik.

Sementara itu, revolusi blockchain berhasil mendesentralisasi lapisan nilai (money layer) dan konsensus, namun belum berhasil mendesentralisasi lapisan data dan komputasi. Mayoritas aplikasi Web3 tetap bergantung pada infrastruktur cloud tradisional untuk penyimpanan, API, dan eksekusi program. Dengan demikian, arsitektur Web3 saat ini masih bergantung pada fondasi Web2.

Di sisi lain, pendekatan desentralisasi penuh tanpa batas kepatuhan menghadapi hambatan adopsi institusional dan risiko hukum. Banyak negara menuntut mekanisme moderasi dan kepatuhan terhadap hukum nasional, terutama untuk layanan publik, tanpa memberikan kendali langsung atas data privat pengguna.

Kesenjangan ini menciptakan ruang desain baru.

DSDN (Distributed Semi-Decentralized Network) diusulkan sebagai infrastruktur data dan komputasi yang:

- Memisahkan secara tegas data plane, compute plane, dan governance plane.
- Mengadopsi prinsip verifiable-by-design, bukan trust-based governance.
- Menggabungkan partisipasi publik permissionless dengan lapisan kepatuhan terbatas yang terverifikasi identitasnya.
- Menyediakan mekanisme replikasi multi-zona, eksekusi terisolasi, dan audit deterministik melalui Data Availability layer.

Tujuan DSDN bukan sekadar menjadi jaringan penyimpanan atau komputasi alternatif, melainkan menjadi infrastruktur fundamental yang mampu menopang sistem berskala besar — mulai dari pendidikan, arsip nasional, komunikasi privat, hingga komputasi AI — tanpa menciptakan konsentrasi kekuasaan baru.

DSDN tidak mengklaim sebagai sistem sepenuhnya trustless atau kebal sensor. Sebaliknya, DSDN secara eksplisit mendefinisikan batas kekuasaan, pembatasan teknis, serta mekanisme audit agar setiap tindakan dapat diverifikasi, dicatat, dan ditantang.

Dengan pendekatan ini, DSDN berupaya menjembatani dua ekstrem: sentralisasi penuh dan desentralisasi tanpa batas, melalui arsitektur yang terukur, dapat diaudit, dan siap diadopsi secara luas.

2. Ringkasan Sistem

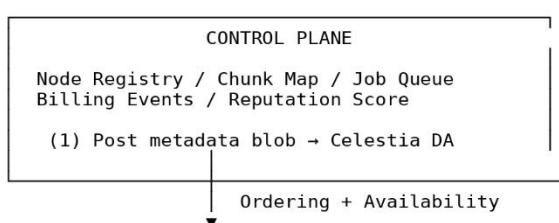
DSDN terdiri dari tiga plane yang saling melengkapi:

1. Control plane — Metadata control plane (node registry, chunk map, job queue, billing events, reputation score) di-post sebagai blob ke Data Availability layer (Celestia) untuk ordering dan availability. Setiap node kemudian mengkonsumsi blob dari Celestia dan membangun state lokal melalui replicated state machine (Tendermint-lite atau log-sink KV store seperti RocksDB/LMDB). Celestia tidak menyimpan state, hanya menjamin ordering dan availability dari blob. State yang sama direkonstruksi secara deterministik oleh setiap node dari urutan blob yang identik.

2. Data dan compute plane dioperasikan oleh full node. Node menyimpan data terpotong dalam chunk yang direplikasi tiga kali di zona berbeda dan mengeksekusi program dalam sandbox.

3. Governance dan compliance plane, dijalankan oleh validator. Pusat governance dan dijalankan bisa sampai 100 – 150 validator, bisa perusahaan, bisa organisasi, bisa lembaga. 100 – 150 ini minimal harus stake 50 ribu token, 20% nya bisa orang yang stake dengan minimal 100 ribu token, menggunakan mekanisme DPoS + quadratic voting dari holder token biasa, dan validator harus mempublish identitas.

Seperti ini ilustrasinya:



2A. Model Kepercayaan dan Asumsi Keamanan

DSDN secara eksplisit mendefinisikan batas kepercayaan (trust boundaries) antar komponen sistem. Tujuan utama dari desain ini adalah meminimalkan entitas yang harus dipercaya dan memastikan bahwa setiap asumsi kepercayaan dapat diverifikasi secara teknis.

Asumsi kepercayaan utama:

1. Celestia Data Availability layer diasumsikan jujur dalam menjamin *ordering* dan *availability* blob, namun tidak dipercaya untuk menyimpan atau mengeksekusi state.
2. Node penyedia storage dan compute diasumsikan tidak dipercaya secara default, dan perilaku mereka diverifikasi melalui replikasi, kuorum, serta mekanisme slashing.
3. Validator diasumsikan tidak dipercaya untuk menjaga privasi data, karena secara teknis tidak memiliki akses terhadap data terenkripsi maupun kunci dekripsi.
4. Coordinator tidak dipercaya sebagai sumber kebenaran; ia hanya bertindak sebagai *stateless scheduler* yang seluruh keputusannya dapat direkonstruksi dan diaudit melalui log Data Availability.

Dengan model ini, kegagalan atau perilaku jahat dari satu atau beberapa komponen tidak dapat mengubah kebenaran state jaringan secara sepihak, melainkan hanya dapat menyebabkan degradasi layanan sementara yang dapat dideteksi dan diaudit.

3. Model Jaringan dan Peran

DSDN mendukung dua kelas node agar partisipasi publik lebih mudah dan biaya operasional lebih rendah, tanpa mengorbankan kualitas layanan untuk beban kerja berskala besar.

Full Node Reguler

Menyimpan data dalam kapasitas terbatas, menjalankan program ringan, dan melayani permintaan pengguna berskala kecil-menengah. Node ini dapat dijalankan menggunakan perangkat kelas rumahan atau server kecil, sehingga partisipasi komunitas menjadi lebih terjangkau. Full node reguler tetap menerima reward berdasarkan kontribusi aktual (storage, compute, bandwidth), namun kapasitas yang lebih kecil membuat total reward yang diterima cenderung lebih rendah.

Stake requirement: Full node reguler wajib melakukan stake minimal 500 \$NUSA untuk mendaftar ke jaringan. Stake ini berfungsi sebagai sybil-resistance dan jaminan perilaku jujur. Stake akan di-slash jika node terbukti melakukan pelanggaran seperti menyajikan data korup atau tidak responsif dalam periode extended.

Full Node Kelas Data Center

Menyediakan kapasitas jauh lebih besar untuk penyimpanan, bandwidth tinggi, dan komputasi intensif termasuk beban GPU. Node kelas ini menjadi pilihan utama scheduler ketika pekerjaan memerlukan reliabilitas, throughput, atau SLA tinggi. Reward tetap mengikuti mekanisme umum DSDN (70% fee storage/compute), tetapi karena volume pekerjaan lebih tinggi, total reward yang diterima lebih besar.

Stake requirement: Full node kelas data center wajib stake minimal 5.000 \$NUSA. Stake lebih besar mencerminkan kapasitas dan tanggung jawab lebih tinggi, serta memberikan sybil-resistance yang proporsional dengan resource yang diklaim.

Anti-Self-Dealing

Untuk mencegah manipulasi reward, DSDN menerapkan aturan anti-self-dealing: node tidak menerima reward dari workload yang di-submit oleh wallet address yang sama atau wallet address yang terafiliasi (ditentukan melalui on-chain graph analysis sederhana). Jika terdeteksi self-dealing, reward dialihkan ke treasury dan node menerima warning. Pelanggaran berulang mengakibatkan slashing stake.

Validator Node

Validator menjalankan governance & compliance plane. Validator hanya memeriksa metadata aplikasi publik dan **tidak dapat mengakses data privat yang terenkripsi end-to-end**. Validator dapat berperan ganda sebagai node kelas data center, namun tetap harus memenuhi persyaratan staking dan kepatuhan.

Selain validator, pemegang token publik (non-validator) juga memiliki peran dalam governance melalui mekanisme Quadratic Voting (QV). Dalam skema ini, setiap akun dapat mengalokasikan suara berdasarkan akar kuadrat dari jumlah token yang di-stake, sehingga suara yang dimiliki seorang pemegang token dihitung sebagai $\sqrt{\text{stake_user}}$. Pendekatan ini menyeimbangkan kekuasaan antara pemegang token besar dan kecil: whale tidak dapat mendominasi keputusan hanya karena memiliki token besar, namun pengguna kecil tetap memiliki bobot yang berarti dalam proses governance. QV digunakan untuk proposal berkaitan dengan penyesuaian tarif, Node Cost Index, parameter teknis jaringan, dan kebijakan non-compliance. Untuk tindakan compliance berat—misalnya memblokir endpoint publik—validator tetap menjadi aktor dominan, namun suara komunitas tetap dihitung melalui QV sebagai secondary check agar keputusan tetap seimbang, transparan, dan tidak mudah dimanipulasi oleh satu pihak.

Batasan kewenangan validator:

1. Validator tidak dapat menghapus atau memodifikasi data aktual yang tersimpan di node. Data terenkripsi secara teknis tidak dapat diakses oleh validator.
2. Validator hanya dapat menghapus atau memblokir pointer/endpoint — yaitu referensi metadata yang mengarah ke data atau aplikasi tertentu. Chunk data tetap ada di node sampai expired atau dihapus oleh pemilik.
3. Validator tidak menggantikan peran full node dalam menyediakan storage atau compute.
4. Validator tidak akan langsung mengatur aplikasi. Validator butuh kesepakatan dari validator lain dan sebagian komunitas (tapi tetap dominan validator) atau proses governance untuk melakukan tindakan seperti memblokir aplikasi, menghentikan sesuatu sistem, dan sebagainya.
5. Semua tindakan compliance harus lewat on-chain governance dengan delay 7–30 hari + multisig dari minimal 60% validator + appeal mechanism.
6. Kalau konten benar-benar ilegal (terorisme, CP), biarkan penegak hukum ambil jalur legal off-chain seperti biasa.

Koordinator

Menyimpan metadata global, peta partisi, status node, dan antrian pekerjaan. Koordinator dijalankan oleh beberapa replika yang mengkonsumsi blob dari Celestia DA dan membangun state lokal secara deterministik. Koordinator tidak menyimpan state authoritative sendiri — state direkonstruksi dari log Celestia. Koordinator tetap netral dan tidak terlibat dalam penyimpanan data pengguna.

Koordinator dalam DSDN tidak memiliki kewenangan otoritatif atas state, reward, maupun eksekusi final. Semua keputusan koordinator—termasuk penjadwalan workload dan penandatanganan receipt—harus dapat direkonstruksi ulang secara deterministik oleh node lain berdasarkan urutan event yang dipost ke Data Availability layer.

Untuk mengurangi risiko konsentrasi kekuasaan, DSDN mendukung model *multi-coordinator*, di mana beberapa instance koordinator berjalan secara paralel dan independen. Receipt dianggap valid hanya jika memenuhi kriteria verifikasi on-chain dan konsisten dengan rekonstruksi state dari log Data Availability. Dalam desain lanjutan, tanda tangan receipt dapat ditingkatkan menjadi skema kuorum (misalnya 2 dari 3 koordinator) atau diverifikasi ulang oleh node lain sebagai mekanisme

Minimal Operational Requirement

Blockchain Nusantara secara teknis dapat berjalan dengan hanya 1 validator node untuk konsensus dan finalitas blok. Ini memungkinkan pengujian, bootstrap awal, atau operasi darurat.

DSDN sebagai sistem storage dan compute membutuhkan persyaratan berbeda:

1. Minimal 3 full node di 3 zona berbeda** diperlukan untuk menjamin durability (RF=3) dan memenuhi SLA ketersediaan.
2. Dengan kurang dari 3 node atau 3 zona, sistem masuk ****mode terdegradasi****: data tidak memenuhi target RF=3, SLA tidak dijamin, dan sistem menampilkan warning kepada pengguna.
3. Angka 100-150 validator dan 3+ node adalah target operasional formal untuk production environment, bukan hard requirement untuk sistem bisa berjalan.

Ringkasan: Blockchain bisa hidup dengan 1 validator. DSDN storage/compute butuh minimal 1 node untuk hidup namun butuh node di 3 zona untuk durability penuh.

Fase Partisipasi Node Awal (Bootstrap Phase)

Pada fase awal jaringan ketika **Replication Factor (RF) = 3**, partisipasi node dan validator bersifat **bootstrap-oriented**, bukan profit-oriented. Node dan validator pada fase ini dioperasikan oleh pihak yang dipercaya (foundation, mitra awal, atau operator internal) dengan tujuan utama **menyediakan stabilitas sistem dan akuisisi pengguna awal**, bukan untuk memaksimalkan pendapatan.

Pada fase ini:

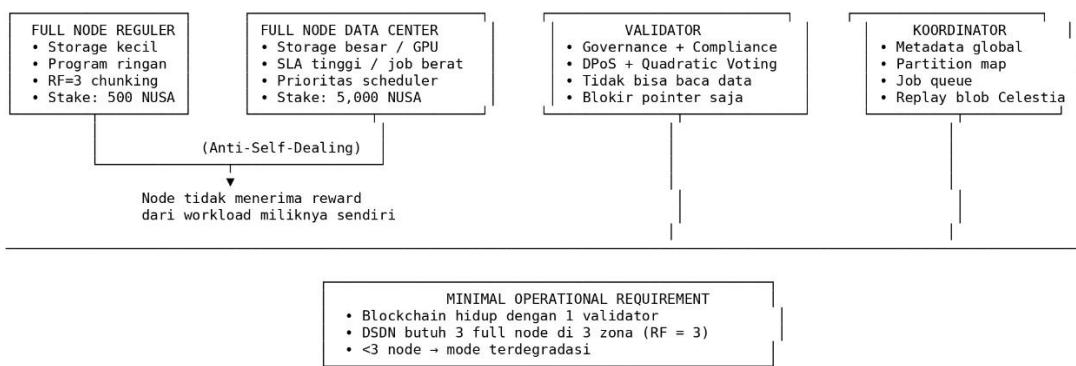
Biaya layanan ditetapkan jauh di bawah tarif pasar normal.

Reward ekonomi untuk node bersifat minimal atau bersubsidi.

Tujuan utama adalah validasi teknis sistem, akuisisi pengguna, dan pembentukan demand riil.

Seiring dengan bertambahnya jumlah node dan zona ($RF > 3$), jaringan secara bertahap bertransisi menuju **mode ekonomi normal**, di mana node publik mulai beroperasi dengan insentif ekonomi penuh.

Ilustrasi:



4. Model Data dan Replikasi

Objek aplikasi dipotong menjadi chunk berukuran 16–64 MiB dan diidentifikasi melalui hash. Penempatan chunk memakai consistent hashing dengan virtual node yang mengenali dua kelas kapasitas: node reguler dan node data center. Target replikasi tetap RF = 3, tetapi sistem memberi prioritas penempatan ke node data center ketika file berukuran besar atau membutuhkan SLA tinggi. Chunk tetap dapat ditempatkan pada node reguler selama memenuhi batas kapasitas dan zona.

Jika jumlah node tidak cukup untuk mencapai tiga replika di tiga zona berbeda, jaringan masuk mode terdegradasi dengan RF lebih rendah dan melakukan self-healing ketika kapasitas kembali tersedia, tanpa memandang kelas node. ****Mode terdegradasi harus dihindari untuk production workload karena tidak memenuhi durability guarantee.****

Konsistensi. Operasi tulis dan baca memakai kuorum replika, misalnya tulis pada dua dari tiga dan baca pada dua dari tiga. Pada dataset masif, erasure coding 6+3 dapat dipakai untuk menurunkan biaya pada node kapasitas besar (biasanya data center).

Ketersediaan. Distribusi replika tetap menjaga tiga domain kegagalan berbeda. Kombinasi node reguler dan node data center dalam satu kota tetap dihitung sebagai satu domain kegagalan; domain harus benar-benar berbeda secara fisik untuk mendekati asumsi independensi kegagalan p.

Privasi dan kunci. Semua objek dienkripsi di sisi klien atau saat unggah. Kunci dikelola oleh pemilik aplikasi, dengan pemulihan memakai Shamir Secret Sharing 2-dari-3 pada pihak yang dipilih pengguna. Node, baik reguler maupun data center, tidak pernah memiliki akses ke kunci dekripsi. ****Validator juga tidak memiliki akses ke kunci dekripsi dan tidak dapat membaca isi data terenkripsi.****

****Konsistensi.**** Operasi tulis dan baca menggunakan kuorum pada tiga replika, misalnya tulis pada dua dari tiga, baca dari dua dari tiga. ****Erasure coding**** seperti skema 6+3 dapat dipakai untuk dataset besar agar mengurangi biaya penyimpanan tanpa mengorbankan daya tahan.

Ketersediaan. Jika probabilitas kegagalan satu zona adalah $*p*$ dan dianggap independen, probabilitas kehilangan seluruh data adalah paling buruk $*p^3*$, sehingga ketersediaan data mendekati $1 - *p^3*$. Zona harus diisolasi secara fisik untuk mendekati asumsi independensi.

Privasi dan kunci. Data aplikasi dienkripsi di sisi klien atau pada saat unggah. Kunci dikelola pemilik aplikasi. Pemulihan kunci memanfaatkan Shamir Secret Sharing dengan skema 2-dari-3 pada pihak tepercaya yang dipilih pemilik.

Pemilik data dapat melakukan penghapusan objek melalui mekanisme User-Controlled Delete. Ketika pengguna meminta penghapusan, sistem menciptakan entri chunk-key blacklist yang dipublikasikan sebagai metadata pada control plane. Node yang menyimpan chunk tersebut akan menandai chunk sebagai "expunged" dan menjadwalkan garbage collection lokal yang menghapus data fisiknya dalam periode 7–30 hari, bergantung pada kelas node. Karena data terenkripsi tidak dapat dibuka oleh operator node, proses ini sepenuhnya berbasis metadata dan tidak melibatkan pemeriksaan isi data. Dengan pendekatan ini, DSDN mendukung hak pengguna untuk menghapus data sambil tetap mempertahankan arsitektur content-addressed yang konsisten.

5. Eksekusi Program

DSDN menjalankan empat kelas beban kerja: situs statis, layanan web stateful, pekerjaan batch, dan proses panjang seperti node blockchain atau pelatihan AI. Pemilihan node untuk menjalankan workload memperhitungkan kelas node serta kebutuhan komputasi.

Isolasi. Program Rust dikompilasi ke WASM/WASI jika memungkinkan, sementara Python dan native Rust berjalan di microVM berbasis Firecracker atau teknologi sejenis. Pada node reguler, batasan cgroups lebih ketat untuk menjaga stabilitas. Node data center dapat menjalankan workload besar, termasuk komputasi GPU dengan time slicing atau MIG.

Hibrida. Pengguna dapat membagi sebagian komputasi ke perangkat mereka sendiri, sedangkan sisanya dijalankan di node reguler atau data center sesuai kebutuhan. Agen klien menyesuaikan mode sinkronisasi parameter (parameter server atau federated averaging) berdasarkan kelas node yang terlibat. Node reguler cocok untuk tugas ringan dan inferensi kecil, sedangkan node data center menangani inferensi besar dan training skala menengah.

Isolasi. Program Rust dikompilasi ke WASM/WASI bila memungkinkan, sedangkan Python dan native Rust berjalan di microVM berbasis Firecracker atau teknologi serupa. Sumber daya dikontrol oleh cgroups. Beban GPU menggunakan penjadwalan time slicing atau MIG jika tersedia.

Hibrida. Pengguna dapat membagi komputasi antara perangkat sendiri dan jaringan. Agen klien menjalankan sebagian komputasi lokal dan mensinkronkan parameter melalui skema parameter server atau federated averaging.

6. Penjadwalan dan Penempatan

Penjadwalan mempertimbangkan kapasitas, kelas node, dan kedekatan data. Skor node didefinisikan sebagai:

$$S = w_1 \cdot \text{CPU_free} + w_2 \cdot \text{RAM_free} + w_3 \cdot \text{GPU_free} + w_4 \cdot (1/\text{latensi}) - w_5 \cdot \text{IO_pressure} + w_6 \cdot \text{class_weight}$$

Di mana:

`class_weight` memberi bobot tambahan pada node kelas data center untuk workload berat (misalnya 1.0 untuk data center, 0.3 untuk node reguler).

Untuk workload ringan atau personal, bobot tersebut dibuat netral agar node reguler tetap ikut digunakan.

Scheduler memilih node dengan skor tertinggi dan tetap menetapkan warm standby untuk failover cepat. Replikasi chunk mengikuti consistent hashing yang sudah mempertimbangkan kapasitas sisa serta proporsi node reguler dan data center.

Migrasi dinamis dapat terjadi ketika node reguler mendekati batas kapasitas. Dalam kasus tersebut, replika tambahan dipindahkan ke node data center melalui proses replikasi malas.

Anti-self-dealing dalam scheduling: Scheduler tidak akan menempatkan workload pada node yang dimiliki oleh wallet address yang sama dengan submitter. Hal ini diverifikasi melalui on-chain ownership registry.

Verifiability dan Audit Receipt

Setiap receipt yang dihasilkan dalam DSDN dipublikasikan dalam bentuk hash ke Data Availability layer, memungkinkan node lain dan auditor independen untuk memverifikasi keberadaan, urutan, dan konsistensi receipt terhadap eksekusi workload yang diklaim. Apabila terjadi ketidaksesuaian—seperti penundaan, penghilangan receipt, atau manipulasi metadata—node yang dirugikan dapat

menyajikan bukti berbasis log Data Availability untuk menantang keabsahan receipt tersebut melalui mekanisme dispute on-chain.

Dengan pendekatan ini, coordinator tidak berfungsi sebagai notaris terpercaya, melainkan sebagai fasilitator yang seluruh tindakannya dapat diverifikasi dan, bila perlu, disanggah.

7. Keamanan dan Privasi

Identitas node menggunakan kunci Ed25519. Antar node terhubung melalui WireGuard atau QUIC dengan mTLS. Artefak program ditandatangani dan dilengkapi SBOM. Sandbox memperketat akses melalui seccomp, AppArmor, pengguna non-root, dan file system hanya baca bila mungkin. OIDC dipakai untuk otentikasi pengguna dengan token berumur pendek. Untuk beban kerja sensitif, enclave seperti SGX atau SEV-SNP dapat diaktifkan.

Sybil-resistance: Setiap node wajib stake token sesuai kelasnya (500 \$NUSA untuk reguler, 5.000 \$NUSA untuk data center). Stake di-slash jika node berperilaku buruk. Mekanisme ini mencegah serangan sybil di mana satu entitas mendaftarkan banyak node palsu untuk memanipulasi reward atau voting.

Slashing terhadap node diterapkan secara bertingkat sesuai tingkat pelanggaran. (1) Liveness failure—node tidak responsif atau gagal melayani chunk selama lebih dari 12 jam—mengakibatkan slashing ringan sebesar 0.5% dari stake. (2) Data corruption—chunk hilang atau rusak pada dua verifikasi kuorum berturut-turut—menghasilkan slashing 5% serta periode cooldown 14 hari sebelum node dapat mendaftar ulang. (3) Repeated malicious behavior—seperti korupsi data berulang atau manipulasi reward—mengakibatkan force unbond dan larangan menjadi node selama 30 hari. Mekanisme ini menciptakan insentif kuat untuk menjaga integritas data dan kualitas layanan tanpa menghukum partisipasi publik secara berlebihan.

Dalam konteks regulasi Indonesia yang ketat terhadap penyimpanan konten ilegal, DSDN menerapkan model keamanan di mana node publik tidak dapat dimintai pertanggungjawaban atas data terenkripsi yang mereka simpan. Karena data tidak

dapat didekripsi tanpa kunci milik pemilik aplikasi, operator node secara teknis dan hukum tidak dapat mengetahui isi data. Sistem hanya memperlihatkan pointer publik kepada validator; jika pointer diblokir, akses aplikasi dinonaktifkan tanpa menghapus data aktual, sehingga jalur hukum tetap mengikuti proses biasa melalui aparat penegak hukum dan bukan melalui operator node. Pendekatan ini memastikan node publik aman secara hukum, sejalan dengan UU ITE, Peraturan Menteri Kominfo, dan prinsip safe harbor penyedia infrastruktur.

8. Moderasi, Validator, dan Kepatuhan

Validator menerapkan aturan deklaratif terhadap aplikasi publik. Pipeline otomatis memeriksa hash daftar terlarang, tanda tangan malware, serta anomali trafik. **Tindakan validator terbatas pada penghapusan atau pemblokiran pointer/endpoint, bukan data aktual.** Semua peristiwa dicatat pada log WORM yang dapat diaudit.

Untuk aplikasi privat, validator tidak melihat isi data, hanya metadata minimum dan jejak hash. **Validator secara teknis tidak dapat mengakses data terenkripsi karena tidak memiliki kunci dekripsi.** Bukti tanpa membuka data dapat ditambahkan kemudian melalui teknik zero-knowledge yang praktis.

Validator dalam DSDN tidak diposisikan sebagai pihak yang “dipercaya” untuk menjaga kebenaran data atau integritas pengguna. Sebaliknya, validator hanya menjalankan fungsi kepatuhan terbatas terhadap layanan publik, dengan kewenangan yang secara sengaja dipersempit dan diawasi melalui mekanisme on-chain. Setiap tindakan validator bersifat reversible secara prosedural, tunduk pada delay eksekusi, dan dapat diaudit oleh publik. Dengan desain ini, DSDN menghindari model kepercayaan berbasis aktor dan menggantinya dengan kepercayaan berbasis pembatasan teknis dan transparansi.

Klarifikasi kewenangan:

1. Validator dapat: memblokir endpoint publik, menghapus pointer metadata, menandai aplikasi untuk review.
2. Validator tidak dapat: membaca data terenkripsi, menghapus chunk dari node, mengakses kunci pengguna.

Agar kewenangan validator tidak disalahgunakan, sistem mendefinisikan Rules of Engagement (RoE) untuk tindakan moderasi. Validator hanya dapat mengusulkan pemblokiran pointer apabila aplikasi atau endpoint publik memenuhi kategori tertentu: (1) konten kriminal berat seperti terorisme dan eksploitasi anak, (2) malware aktif atau penipuan terstruktur, (3) pelanggaran hukum telekomunikasi yang jelas dan terverifikasi. Konten yang bersifat politis, kritik sosial, ekspresi pribadi, atau informasi yang tidak melanggar hukum tidak dapat dijadikan dasar tindakan validator. Untuk kategori abu-abu, keputusan harus melewati multi-stage review yang mencakup voting validator (minimal 60%), Quadratic Voting dari komunitas, dan delay 7-30 hari sebagai jaminan transparansi. Dengan pendekatan ini, DSDN menjaga keseimbangan antara kepatuhan hukum, kebebasan berekspresi, dan perlindungan terhadap penyalahgunaan kekuasaan.

DSDN menggunakan model governance hybrid yang menggabungkan diskusi off-chain dan pengambilan keputusan on-chain. Proposal diawali melalui forum publik atau repository diskusi yang memungkinkan analisis teknis, argumentasi hukum, dan kontribusi komunitas. Setelah melalui tahap draft, proposal dipublikasikan on-chain sebagai governance item dan divoting menggunakan dua lapisan: validator vote dan Quadratic Voting komunitas. Model ini meniru praktik jaringan besar seperti Cosmos, Optimism, dan Arbitrum, di mana deliberasi kompleks lebih efisien dilakukan di luar rantai, sementara keputusan akhir tetap final dan transparan melalui on-chain governance.

8.A Model Governance Bertahap (Progressive Governance Model)

Untuk memastikan bahwa DSDN dapat beroperasi secara stabil sejak fase awal peluncuran hingga mencapai tingkat desentralisasi penuh, sistem governance DSDN dirancang menggunakan pendekatan **bertahap (progressive governance)**. Dalam model ini, **kelangsungan operasi sistem (liveness)** tidak bergantung pada keberadaan maupun jumlah validator governance, sementara distribusi kewenangan dilakukan secara gradual seiring dengan pertumbuhan dan pematangan jaringan.

Prinsip utama dari desain ini adalah bahwa **data plane, compute plane, dan blockchain Nusantara harus tetap dapat beroperasi secara otomatis, deterministik, dan independen**, bahkan ketika governance belum aktif atau belum berjalan secara penuh.

Fase 0 — Foundational / Pre-Governance Mode

Pada fase awal peluncuran jaringan, ketika jumlah validator masih berada di bawah ambang minimum atau belum tersedia sama sekali, DSDN beroperasi dalam **Pre-Governance Mode**.

Karakteristik utama:

Tidak terdapat governance publik, mekanisme voting, maupun proposal on-chain.

Seluruh parameter sistem bersifat statis dan ditetapkan melalui konfigurasi awal protokol.

Operasi storage, compute, scheduling, billing, dan replikasi berjalan sepenuhnya secara otomatis berdasarkan aturan teknis yang telah ditentukan.

Kelangsungan sistem tidak bergantung pada quorum governance maupun persetujuan validator.

Pada fase ini, kewenangan operasional terbatas dipegang oleh **Foundation Key** (atau **Founder Authority**) dengan ruang lingkup yang secara eksplisit dibatasi, meliputi:

Penerapan pembaruan parameter teknis.

Tindakan darurat terbatas, seperti penangguhan sementara endpoint publik yang bermasalah.

Penerapan patch keamanan atau perbaikan bug kritis.

Foundation Key tidak memiliki akses terhadap data terenkripsi pengguna, tidak dapat menghapus chunk data, serta tidak memiliki kewenangan untuk memindahkan atau menarik dana pengguna. Seluruh tindakan Foundation dicatat secara transparan sebagai event on-chain dan dapat diaudit.

Fase 1 – Bootstrap / Semi-Governance Mode

Ketika jaringan mencapai **minimal tiga validator aktif** ($RF\ validator \geq 3$), DSDN memasuki **Bootstrap Governance Mode**. Pada fase ini, modul governance telah aktif secara teknis, namun belum memiliki kewenangan penuh dalam mengeksekusi keputusan protokol.

Karakteristik utama:

Validator dapat mengajukan proposal dan melakukan voting, namun hasil voting bersifat **rekомendatif (non-binding)**.

Proposal governance tidak dieksekusi secara otomatis tanpa persetujuan Foundation.

Governance berfungsi sebagai sarana pengujian mekanisme voting, koordinasi antar-validator, serta simulasi kebijakan protokol.

Slashing berbasis governance belum diterapkan secara penuh; hanya mekanisme otomatis seperti kegagalan liveness yang aktif.

Struktur kekuasaan pada fase ini bersifat **semi-terpusat**, dengan pembagian peran sebagai berikut:

Foundation memiliki hak override dan veto terbatas.

Validator berperan sebagai aktor advisory serta penguji stabilitas mekanisme governance.

Komunitas dapat berpartisipasi secara terbatas, misalnya melalui simulasi voting tanpa dampak langsung terhadap eksekusi protokol.

Seluruh tindakan override atau veto yang dilakukan oleh Foundation harus dicatat secara transparan pada blockchain sebagai bagian dari audit trail.

Fase 2 — Transition Governance Mode

Ketika jumlah validator meningkat dan mencapai tingkat stabilitas tertentu (misalnya 50–99 validator aktif), jaringan memasuki **Transition Governance Mode**.

Karakteristik utama:

Proposal governance mulai dieksekusi secara otomatis setelah melewati periode delay yang telah ditentukan.

Mekanisme **Quadratic Voting (QV)** komunitas mulai diperhitungkan sebagai bagian dari proses pengambilan keputusan.

Slashing berbasis governance diterapkan secara terbatas, disertai mekanisme banding (appeal).

Peran Foundation dikurangi secara bertahap dan tidak lagi bersifat absolut.

Pada fase ini, kewenangan didistribusikan secara bertingkat antara Foundation, validator, dan komunitas, dengan tujuan mempersiapkan transisi menuju desentralisasi penuh tanpa mengorbankan stabilitas sistem.

Fase 3 — Full Governance Mode

Ketika jaringan mencapai skala operasional penuh dengan **100–150 validator aktif**, DSDN memasuki **Full Governance Mode**, sebagaimana didefinisikan dalam desain utama whitepaper.

Karakteristik utama:

Tidak terdapat lagi hak veto maupun override dari Foundation.

Seluruh keputusan governance dijalankan sepenuhnya melalui mekanisme on-chain.

Validator menjadi aktor utama dalam pengambilan keputusan, dengan dukungan **Quadratic Voting** dari komunitas.

Seluruh tindakan governance mengikuti **Rules of Engagement (RoE)**, periode delay eksekusi 7–30 hari, mekanisme multisignature validator, serta sistem banding.

Pada fase ini, Foundation berperan sebagai partisipan biasa (misalnya sebagai validator atau kontributor), tanpa kewenangan khusus pada tingkat protokol.

Transisi dan Keamanan Governance

Mode governance DSDN direpresentasikan sebagai variabel status protokol (`governance_mode`) yang memengaruhi perilaku modul governance serta eksekusi kebijakan. Perubahan mode governance hanya dapat terjadi melalui:

Pemenuhan ambang batas teknis yang telah ditentukan, seperti jumlah validator aktif dan stabilitas jaringan, atau

Keputusan governance on-chain pada fase lanjutan.

Dengan pendekatan governance bertahap ini, DSDN menghindari risiko desentralisasi prematur, menjaga stabilitas sistem pada fase awal, serta menyediakan jalur yang jelas, transparan, dan terukur menuju governance terbuka dan desentralisasi penuh.

9. Operasi dan Observabilitas

DSDN mengumpulkan metrik standar untuk CPU, RAM, GPU, I/O, kapasitas, dan kesehatan replikasi. Log dan jejak disekat per tenant. Peringatan diterbitkan ketika kapasitas rendah, replika berkurang dari target, pekerjaan macet, atau node sering

naik turun. Jaringan mempertahankan cadangan kapasitas 20 sampai 30 persen untuk menyerap lonjakan dan kegagalan mendadak.

State reconstruction: Karena state dibangun dari blob Celestia, setiap node dapat melakukan state recovery dengan me-replay blob dari genesis. Hal ini memudahkan debugging dan audit.

10. Skala, Biaya, ekonomi dan Ketersediaan

Dengan replikasi tiga kali, biaya penyimpanan adalah tiga kali ukuran data mentah tanpa erasure coding. Untuk dataset besar, skema 6+3 menurunkan overhead menjadi 1.5 kali dengan daya tahan yang sebanding. Latensi baca menurun ketika data ditempatkan pada tiga zona terdekat dengan anycast DNS. Model biaya dapat dibagi antara operator node komunitas dan fasilitas pusat yang menyediakan jaringan dan penyeimbang lalu lintas.

Ekonomi:

DSDN diciptakan dibarengi dengan blockchain untuk transaksi dan validasi sistem keseluruhan DSDN, dan yang menjalankan node DSDN, akan dibarengi menjalankan blockchain, node akan dibayar dengan token internal DSDN (\$NUSA) dan token akan masuk ke wallet orang pemilik node, agar orang mau jalanin node, blockchain ini bernama Nusantara, menggunakan konsensus proof of stake di node validator DSDN, suplai token DSDN (\$NUSA) max adalah 300 juta

SEGALA AKTIVITAS DI DSDN DIBAYAR MENGGUNAKAN \$NUSA COIN,

SPESIFIKASI BLOCKCHAIN:

1. BERBASIS ACCOUNT BUKAN UTXO SEPERTI BITCOIN
2. HASH UTAMA ADALAH SHA3 512
3. Konsensus -> PoS

4. Db -> LMDB

5. SEMI DECENTRAL KARENA SEMUA ORANG BISA MENJALANKAN NODE DAN SISTEM, TAPI VALIDATOR MILIK PERUSAHAAN BISA MENGONTROL AGAR DSDN TETAP BERETIKA (maksudnya mengontrol adalah menghilangkan konten eksplisit saja melalui penghapusan pointer/endpoint, sisanya diatur oleh komunitas dan governance)

Blockchain Nusantara dirancang dengan throughput menengah untuk mendukung operasi metadata DSDN tanpa menjadi bottleneck. Target awal adalah 500–1.500 transaksi per detik, dengan waktu blok 2–4 detik dan finalitas deterministik kurang dari 5 detik melalui konsensus Proof of Stake. Kapasitas ini lebih dari cukup karena operasi berat seperti upload data, replikasi chunk, dan eksekusi komputasi tidak berjalan di blockchain, melainkan di data plane. Blockchain hanya mencatat transaksi user, staking, governance, event billing, dan pointer metadata, sehingga beban transaksinya relatif stabil dan dapat dikelola tanpa memerlukan skala ekstrem seperti jaringan L1 global.

Control plane architecture: State blockchain dan DSDN tidak disimpan di Celestia. Celestia hanya menyediakan Data Availability layer untuk ordering blob. Setiap node membangun state lokal dengan mengkonsumsi blob secara berurutan dan menjalankan state transition secara deterministik. Pendekatan ini mirip dengan Tendermint-lite atau log-structured state machine.

JIKA INGIN BERAKTIVITAS DI DSDN SEPERTI UPLOAD FILE, MENGENSKRIPSI FILE, JALANKAN RUNTIME, MAKA AKTIVITAS ITU SEMUA AKAN TERHUBUNG KE WALLET USER(ADDRES & PRIVKEY), DAN USER YANG MELAKUKAN AKTIVITAS BAKAL MEMBAYAR GAS FEE, DAN MENANDATANGANI DIGITAL SIGNATURE

Text lengkap:

"Token \$NUSA memiliki suplai maksimal 300 juta token tanpa mekanisme pencetakan baru, sehingga suplai bersifat deflasi dari waktu ke waktu. Distribusi awal dibuat konservatif untuk menjaga kredibilitas proyek: 60% dialokasikan untuk "Node & Validator Reward Pool" yang dilepas bertahap selama dua puluh tahun, 20% untuk Foundation dan riset dengan vesting lima tahun, 10% untuk community grants, 5% untuk hadiah early testnet node, dan 5% untuk kebutuhan likuiditas. Porsi terbesar memang diarahkan kepada node dan validator karena mereka menjalankan seluruh infrastruktur DSDN."

Node mendapatkan pendapatan langsung dari aktivitas pengguna. Setiap operasi seperti mengunggah atau mengunduh data, menyimpan file, menjalankan runtime WASM atau Python, melakukan inferensi AI, atau menjalankan aplikasi jangka panjang akan menghasilkan biaya dalam bentuk \$NUSA. Biaya ini dibagi otomatis: 70% kepada node yang menyediakan resource (storage atau compute), 20% kepada validator sebagai imbalan staking, dan 10% masuk ke treasury untuk audit, pengembangan, dan mekanisme burn. **Catatan: Node tidak menerima reward dari workload yang di-submit oleh wallet address miliknya sendiri (anti-self-dealing).** Dengan pola ini, node tidak bergantung pada reward inflasi, tetapi memperoleh penghasilan dari layanan nyata yang digunakan pengguna.

Tarif yang dicantumkan pada bagian ini merepresentasikan tarif standar jaringan yang berlaku pada fase ekonomi normal.

Pada fase bootstrap jaringan, khususnya ketika Replication Factor (RF) masih berada pada nilai minimum ($RF = 3$), biaya layanan dapat diturunkan secara signifikan dari tarif standar ini untuk mendorong adopsi pengguna awal dan mengurangi hambatan masuk.

Kenaikan biaya menuju tarif standar dilakukan secara bertahap seiring dengan peningkatan kapasitas jaringan, jumlah node, dan tingkat replikasi ($RF > 3$), sehingga transisi ekonomi berlangsung secara stabil tanpa menimbulkan shock biaya bagi pengguna.

Stake requirement untuk node:

- Full node reguler: minimal 500 \$NUSA
- Full node data center: minimal 5.000 \$NUSA
- Validator: minimal 50.000 \$NUSA (seperti tercantum di Bagian 2)

Validator dalam sistem Proof of Stake menerima dua aliran pendapatan: 20% dari seluruh fee aktivitas pengguna dan reward tahunan kecil hingga 1% untuk insentif staking. Validator wajib menaruh stake besar agar jaringan tetap stabil dan tidak mudah diserang.

Untuk menjaga harga token tetap sehat dan biaya tetap wajar bagi pengguna, sistem menggunakan tiga mekanisme stabilisasi. Pertama, sebagian dari treasury dibakar secara berkala untuk menurunkan suplai secara bertahap. Kedua, gas fee bersifat adaptif dan dapat menyesuaikan harga token: ketika harga naik, biaya dasar dapat diturunkan, dan sebaliknya. Ketiga, DSDN menggunakan "Node Cost Index," yaitu modul on-chain yang memantau faktor seperti harga listrik, biaya storage, bandwidth, serta beban jaringan. Index ini mengusulkan penyesuaian fee secara periodik, yang kemudian divoting validator. Dengan cara ini, biaya tetap mengikuti realita ekonomi tanpa membebani pengguna atau merugikan node.

Simulasi sederhana menunjukkan bahwa seorang pengguna yang menyimpan 100 GB data, mengunggah 10 GB, mengunduh 30 GB, dan menjalankan 100 menit CPU serta 10 menit GPU per bulan menghabiskan sekitar 18.9 \$NUSA. Dengan harga token sekitar Rp10.000, biaya bulanan sekitar Rp189.000—jauh lebih rendah dibanding layanan cloud komersial. Node yang melayani pengguna tersebut menerima sekitar 13.23 \$NUSA. Jika satu node memiliki lima puluh pelanggan aktif, total pendapatan bulanannya dapat mencapai sekitar 661 \$NUSA atau sekitar 6.6 juta rupiah, cukup untuk menutup biaya operasi dan memberikan profit yang stabil.

Dengan mekanisme ini, DSDN memiliki token yang benar-benar digunakan untuk layanan, bukan sekadar spekulasi. Biaya tetap ramah pengguna, node mendapat pendapatan yang konsisten, validator memiliki insentif jangka panjang, dan suplai token terjaga. Seluruh struktur ini menciptakan ekosistem yang berkelanjutan tanpa mengandalkan hype."

10.1 Mekanisme Deflasi dan Stabilitas Nilai Token \$NUSA

Token \$NUSA dirancang sebagai token utilitas infrastruktur dengan suplai maksimal tetap dan mekanisme deflasi terkontrol. Tujuan utama desain ini adalah menjaga daya beli token dalam jangka panjang dan melindungi ekosistem DSDN dari tekanan inflasi mata uang fiat.

Deflasi \$NUSA tidak bersifat tetap, melainkan berbasis aktivitas ekonomi jaringan, dengan target deflasi tahunan pada kisaran 3–6%. Mekanisme deflasi ini terdiri dari beberapa lapisan:

1. Pembakaran Treasury Berkala

Sebagian dana treasury yang berasal dari biaya jaringan dibakar secara berkala. Laju pembakaran disesuaikan dengan tingkat aktivitas jaringan dan kondisi ekonomi, sehingga deflasi tetap terkendali.

2. Pembakaran Berbasis Penggunaan

Setiap aktivitas storage dan compute menghasilkan pembakaran tambahan dalam proporsi kecil. Semakin tinggi penggunaan jaringan, semakin besar tekanan deflasi yang terjadi.

3. Slashing sebagai Sink Deflasi

Sebagian token yang di-slash akibat pelanggaran node atau validator dibakar, sementara sisanya dialokasikan ke treasury. Pendekatan ini meningkatkan keamanan sekaligus menambah tekanan deflasi.

4. Penguncian Suplai melalui Staking

Token yang di-stake oleh node, validator, dan delegator terkunci dalam periode tertentu, sehingga menurunkan suplai beredar efektif dan memperkuat stabilitas nilai token.

Dengan pendekatan ini, nilai \$NUSA didukung oleh penggunaan riil jaringan, bukan spekulasi semata, dan dirancang untuk tetap stabil serta tahan terhadap inflasi dalam jangka panjang.

11. Implementasi Referensi

Bahasa utama adalah rust dan Python. Komponen kunci:

* Koordinator (rust) – mengkonsumsi blob dari Celestia dan membangun state lokal via replicated state machine.

- * Penyimpanan objek (rust) dengan content addressing, consistent hashing, dan self-healing.
- * Runtime WASM dan microVM untuk eksekusi aman.
- * Ingress HTTP/HTTP3 dengan rute sadar geografi.
- * Agen klien untuk hybrid compute dan sinkronisasi parameter.
- * State machine (rust) — log-sink KV store untuk state reconstruction dari Celestia blob.

Interface minimum antara komponen diekspos melalui API yang sederhana untuk status aplikasi, penempatan data, dan validasi kebijakan.

-

12. Kesimpulan

DSDN menawarkan jalur praktis untuk jaringan data dan komputasi yang tersebar, dapat diaudit, dan melindungi privasi. Sistem ini mengurangi ketergantungan pada pusat data tunggal, membagi beban listrik, dan membuka partisipasi luas dari masyarakat, dengan tetap memberi ruang untuk penegakan hukum pada layanan publik. Arsitektur ini cukup sederhana untuk diimplementasikan bertahap, namun cukup kuat untuk menopang pendidikan, arsip, komunikasi, dan komputasi AI di Indonesia.

Batasan dan Non-Tujuan Sistem

DSDN tidak mengklaim sebagai sistem yang sepenuhnya trustless atau kebal terhadap seluruh bentuk sensor dan kegagalan. Sistem ini tidak menjanjikan anonimitas absolut, tidak menghilangkan seluruh risiko kolusi, dan tidak menggantikan proses hukum yang berlaku. Sebaliknya, DSDN secara sadar memilih pendekatan *verifiable, auditable, and constrained*, di mana setiap bentuk kekuasaan dibatasi, dicatat, dan dapat ditantang.

Dengan mendefinisikan secara eksplisit apa yang tidak dijanjikan, DSDN bertujuan membangun kepercayaan jangka panjang melalui kejujuran desain, bukan klaim absolut yang sulit diverifikasi.

Lampiran B. Algoritme Ringkas

Penempatan replika: hitung hash objek, pilih tiga node berikutnya pada ring yang berada di zona berbeda dan memiliki kapasitas memadai. Jika kurang dari tiga, tandai terdegradasi dan jadwalkan replikasi ulang saat tersedia.

Self-healing: agen latar memindai objek yang terdegradasi, membuat replika baru hingga kembali ke target RF.

Skor penjadwalan: gunakan rumus pada Bagian 6 dengan bobot yang dikalibrasi. Terapkan preemption untuk menjaga SLO pada kelas Critical.

Anti-self-dealing check: sebelum assign reward, verifikasi bahwa node owner ≠ workload submitter melalui on-chain registry.

-

Lampiran C. Spesifikasi Node Referensi (Revisi Dua Kelas Node)

1. Full Node Reguler (Partisipasi Publik)

Dirancang agar publik dapat ikut berkontribusi tanpa biaya perangkat yang tinggi.

- CPU: 4–8 vCPU
- RAM: 8–32 GiB
- Storage:
 - NVMe 512 GB – 2 TB (hot tier)
 - HDD optional 2–4 TB (warm tier)

- Jaringan: 300 Mbps – 1 Gbps
- GPU: Opsional (jika ada, digunakan untuk compute ringan)
- Power: UPS 5–10 menit (opsional)
- Lokasi: Rumah, kantor kecil, atau server low-cost
- ****Stake: Minimal 500 \$NUSA****

Node reguler tetap berperan dalam replikasi chunk (RF=3), tetapi scheduler akan mengarahkan pekerjaan berat ke node data center.

2. Full Node Kelas Data Center (Kapasitas Tinggi)

Dirancang untuk storage besar, workload kritis, dan compute berat.

- CPU: 32–64 vCPU
- RAM: 128–256 GiB
- Storage:
 - NVMe 4–8 TB untuk hot tier
 - HDD 8–16 TB untuk warm/cold tier
- Jaringan: 10–25 Gbps dengan dua uplink
- GPU: 24–48 GB VRAM (opsional namun direkomendasikan)
- Power: UPS 30 menit + genset lokasi
- Lokasi: Minimal tiga zona berbeda per kota untuk ketersediaan tinggi
- **Stake: Minimal 5.000 \$NUSA**

Node jenis ini akan lebih sering terpilih untuk workload besar, sehingga memperoleh reward lebih besar karena kapasitas dan stabilitasnya.

Lampiran D. Ringkasan Patch v0.5.2

Perubahan dari v0.5.1:

1. ****Control-plane architecture (Bagian 2, 10, 11):**** Diperjelas bahwa Celestia hanya untuk DA/ordering, state dibangun lokal via replicated state machine (Tendermint-lite / log-sink KV store).
2. ****Validator limitation (Bagian 3, 4, 8):**** Ditegaskan bahwa validator hanya dapat menghapus pointer/endpoint, bukan data aktual. Validator tidak dapat mengakses data terenkripsi.
3. ****Sybil-resistance & anti-self-dealing (Bagian 3, 6, 7, 10, Lampiran B, C):****
Ditambahkan stake requirement untuk node (500 \$NUSA reguler, 5.000 \$NUSA data center). Node tidak menerima reward dari workload yang di-submit sendiri.
4. ****Minimal operational requirement (Bagian 3):**** Diperjelas bahwa blockchain bisa hidup dengan 1 validator, tetapi DSDN storage/compute butuh minimal 3 node di 3 zona untuk durability dan SLA.