

Perumusan dan Penyelesaian Persamaan Kesetimbangan dalam Analisis Dinamis pada Bangunan Tinggi Berbasis Website

Hans Christian Herjanto¹, Justinus Andjarwirawan², Henry Novianus Palit³

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

Email : hanschrizt@gmail.com¹, justin@petra.ac.id², hnpalit@petra.ac.id³

ABSTRAK

Aplikasi yang umum digunakan untuk menyelesaikan persamaan analisis dinamis menghasilkan *output* yang kurang akurat karena penggunaan implementasi massa yang kurang tepat. Aplikasi tersebut juga merupakan aplikasi komersial dan memerlukan proses instalasi sebelum dapat digunakan. Penulis membuat aplikasi menggunakan media *website* sehingga mudah diakses dan tidak memerlukan proses instalasi. *Website* dibuat dengan menggunakan implementasi massa yang lebih tepat sehingga *output* menjadi lebih akurat dan dilengkapi dengan simulasi deformasi dan defleksi struktur.

Penulis merancang *website* sehingga *user* dapat memiliki akun untuk membuat *project* yang bersifat *public* atau *private*, serta *admin* untuk mengolah data *member*. Proses *input* dibuat dengan tabel dengan *tab-tab* yaitu *node*, *properties*, *element*, *joint mass*, *joint loads*, *ground acceleration*, *process* sehingga *user* mudah untuk melakukan *input* data. Proses perhitungan analisis dinamis dilakukan dengan mengolah data *input* menggunakan metode *Newmark* seperti pembuatan matriks kekakuan, massa dan redaman, inialisasi perpindahan awal, kecepatan dan percepatan, perhitungan konstanta integrasi, pembuatan matriks kekakuan element, perhitungan perpindahan waktu, perhitungan perpindahan, kecepatan dan percepatan.

Pembuatan *website* analisis dinamis menghasilkan *output* seperti matriks pada setiap element, matriks kekakuan struktur, matriks massa struktur, dan vektor perpindahan yang sesuai dengan solusi eksak. Hasil vektor perpindahan digunakan untuk menampilkan simulasi deformasi dan defleksi struktur pada setiap *interval* waktu sehingga terlihat perubahan struktur yang terjadi dari awal hingga akhir.

Kata Kunci: Analisis Dinamis, Metode Integrasi Langsung, Metode Newmark

ABSTRACT

Applications commonly used to solve the equations of dynamic analysis produce output that is less accurate because the use of mass implementation is less precise. Most of the applications are commercial and require the installation process before they can be used. The authors developed an application that uses website as the user interface, so that it can be accessed easily and does not require any installation process. The Website was created by using a more precise mass implementation so that the output becomes more accurate with simulated deformation and structure deflection.

The Authors designed the website so that a user can have an account to make a project that is public or private, and an

admin can manage the member data. Input process data are entered into tables with tabs such as node, properties, elements, joint mass, joint loads, ground acceleration, process so that the user can perform data input easily. Dynamic analysis calculation is performed by processing the input data using the Newmark method involving the creation of stiffness, mass, damping matrices the initialization of displacement, velocity, and acceleration the calculation of integration constants, the creation of element stiffness matrix, the calculation of displacement with time dependent, the calculation of displacement, velocity and acceleration.

This dynamic analysis website generates outputs like all mentioned matrices in each element, structure stiffness matrix, structure mass matrix, and displacement vector that conform with the exact solution. Results of displacement vector are used to display the simulation of deformation and structure deflection at each time interval so the the strukturak changes are observable from the beginning to the end.

Keywords: Dynamic Analysis, Direct Integration Method, Newmark Method

1. PENDAHULUAN

Desain gedung didasarkan pada dua analisis yaitu analisis statis dan analisis dinamis yang saling berkaitan. Analisis statis merupakan analisis dimana gedung didesain dengan persamaan kesetimbangan saat gedung diam (tidak mengalami gaya eksternal yang berubah-ubah), sedangkan analisis dinamis merupakan analisis dimana gedung didesain secara khusus agar gedung dapat menahan gaya eksternal yang bekerja pada waktu/periode tertentu.

Penyelesaian persamaan kesetimbangan diperlukan untuk memberikan solusi analisis dinamis pada bangunan tinggi, sehingga dapat diketahui gaya dinamis yang harus ditahan oleh gedung. Terdapat aplikasi yang dapat digunakan untuk membantu menentukan penyelesaian persamaan analisis dinamis tersebut, namun aplikasi yang ada menghasilkan *output* yang kurang akurat karena penggunaan implementasi massa yang kurang tepat. Aplikasi yang ada juga perlu di *install* terlebih dahulu pada suatu komputer sebelum dapat digunakan untuk mendesain gedung.

Adanya aplikasi berbasis *website* untuk penyelesaian persamaan kesetimbangan analisis dinamis diharapkan dapat meningkatkan efisiensi waktu karena aplikasi ini dapat dengan mudah diakses tanpa perlu di *install* terlebih dahulu dan dapat diakses tidak hanya melalui komputer sehingga mudah untuk digunakan dimana saja. Aplikasi juga dibuat menggunakan implementasi

massa yang lebih tepat sehingga dihasilkan *output* analisis dinamis yang akurat dan dapat digunakan dalam mendesain gedung.

Pembuatan *website* penyelesaian persamaan kesetimbangan dalam analisis dinamis pada bangunan tinggi ini bekerja sama dengan narasumber yaitu mahasiswa program studi teknik sipil Universitas Kristen Petra yang bernama Kevin Winata (21411023) dan dosen pembimbing skripsi dari Kevin Winata, yaitu Bpk. Wong Foek Tjong, Ph.D. Topik skripsi ini diangkat dari ide yang diberikan oleh Bpk. Wong Foek Tjong, Ph.D. *Website* ini mengembangkan *website* penyelesaian persamaan kesetimbangan dalam analisis statis pada bangunan tinggi yaitu <http://ta29.petra.ac.id>.

2. LANDASAN TEORI

2.1 Persamaan Kesetimbangan Analisis

Dinamis

Persamaan kesetimbangan dari respon dinamis linear pada sebuah sistem struktur adalah

$$\mathbf{M} \ddot{\mathbf{U}} + \mathbf{C} \dot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F}(t) \quad (1)$$

Pada persamaan 1 \mathbf{M} adalah matriks massa struktur, $\ddot{\mathbf{U}}$ adalah vektor atau matriks kolom percepatan titik-titik *node*, \mathbf{C} adalah matriks redaman struktur, $\dot{\mathbf{U}}$ adalah vektor atau matriks kolom kecepatan titik-titik *node*, \mathbf{K} adalah matriks kekakuan struktur, \mathbf{U} adalah vektor atau matriks kolom perpindahan titik-titik nodal, dan $\mathbf{F}(t)$ adalah vektor gaya pada titik-titik *node* yang memiliki ketergantungan terhadap waktu. Analisis dinamis dapat dilakukan dengan dua cara, yang pertama dapat menggunakan *modal analysis* dan yang kedua dapat menggunakan metode integrasi langsung. Tahapan-tahapan penyelesaian analisis dinamis akan dilakukan dengan menggunakan metode integrasi langsung [1].

2.2 Metode Integrasi Langsung

Persamaan 1 adalah persamaan integrasi langsung yang diintegrasikan menggunakan prosedur numerik yang bertahap. Metode integrasi langsung ini didasari oleh dua pemikiran, yaitu digunakan untuk memenuhi persamaan 1 pada seluruh waktu atau t yang terbagi dalam interval waktu atau Δt , serta variasi perpindahan, kecepatan dan percepatan pada setiap interval waktu atau Δt yang berupa asumsi. Variasi dari asumsi interval waktu atau Δt serta perpindahan, kecepatan dan percepatan ini dapat menentukan akurasi, stabilitas serta prosedur penyelesaian.

Vektor perpindahan, kecepatan, dan percepatan diasumsikan pada kondisi awal ($t = 0$), dapat dituliskan sebagai ${}^0\mathbf{U}$, ${}^0\dot{\mathbf{U}}$, dan ${}^0\ddot{\mathbf{U}}$, masing-masing diketahui sehingga solusi persamaan 1 dapat terpenuhi untuk $t=0$ hingga $t=T$. Pada solusi jangka waktu di bawah pertimbangan, T dapat dibagi menjadi n interval waktu yang jumlahnya sama yaitu Δt , maka dapat ditulis bahwa $\Delta t = T/n$, dan skema integrasi yang digunakan menetapkan solusi perkiraan pada waktu Δt , $2\Delta t$, $3\Delta t$, ..., t , $t + \Delta t$, ..., T . Berbagai algoritma dapat digunakan untuk menghitung persamaan pada setiap interval waktu.

2.2.1 Metode Newmark

Skema integrasi *Newmark* dapat dikatakan sebagai perkembangan dari metode percepatan linear. Terdapat beberapa asumsi persamaan seperti

$${}^{t+\Delta t}\ddot{\mathbf{U}} = {}^t\ddot{\mathbf{U}} + [(1 - \delta) {}^t\ddot{\mathbf{U}} + \delta {}^{t+\Delta t}\ddot{\mathbf{U}}] \Delta t \quad (2)$$

$${}^{t+\Delta t}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + {}^t\ddot{\mathbf{U}} \Delta t + \left[\left(\frac{1}{2} - \alpha \right) {}^t\dot{\mathbf{U}} + \alpha {}^{t+\Delta t}\dot{\mathbf{U}} \right] \Delta t \quad (3)$$

Untuk α dan θ merupakan parameter yang telah ditentukan untuk mendapatkan integrasi yang stabil dan juga akurat. Pada saat $\delta = \frac{1}{2}$ dan $\alpha = \frac{1}{6}$, maka hubungan persamaan 2 dan persamaan 3 sesuai dengan metode percepatan linear seperti didapatkan dengan $\theta = 1$ pada metode Wilson θ . Tahapan solusi penyelesaian analisis dinamis menggunakan metode *Newmark* adalah sebagai berikut.

A. Perhitungan awal:

1. Bentuk matriks kekakuan \mathbf{K} , matriks massa \mathbf{M} , serta matriks redaman \mathbf{C} .
2. Tentukan inialisasi ${}^0\mathbf{U}$, ${}^0\dot{\mathbf{U}}$, dan ${}^0\ddot{\mathbf{U}}$.

3. Tentukan tahapan waktu atau Δt dan tentukan parameter α , δ , kemudian hitung konstanta integrasi:

$$\delta \geq 0.50; \quad \alpha \geq 0.25(0.5 + \delta)^2$$

$$a_0 = \frac{1}{\alpha \Delta t^2}; \quad a_1 = \frac{\delta}{\alpha \Delta t}; \quad a_2 = \frac{1}{\alpha \Delta t}; \quad a_3 = \frac{1}{2\alpha} - 1;$$

$$a_4 = \frac{\delta}{\alpha} - 1; \quad a_5 = \frac{\Delta t}{2} \left(\frac{\delta}{\alpha} - 2 \right); \quad a_6 = \Delta t(1 - \delta); \quad a_7 = \delta \Delta t$$

4. Bentuk matriks kekakuan elemen $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{K} + a_0\mathbf{M} + a_1\mathbf{C}$
5. *Triangularize* $\hat{\mathbf{K}}$: $\hat{\mathbf{K}} = \mathbf{LDL}^T$

B. Untuk setiap tahapan waktu:

1. Hitung beban efektif pada waktu $t + \Delta t$:

$${}^{t+\Delta t}\hat{\mathbf{R}} = {}^{t+\Delta t}\mathbf{R} + \mathbf{M}(a_0 {}^t\mathbf{U} + a_2 {}^t\dot{\mathbf{U}} + a_3 {}^t\ddot{\mathbf{U}}) + \mathbf{C}(a_1 {}^t\mathbf{U} + a_2 {}^t\dot{\mathbf{U}} + a_5 {}^t\ddot{\mathbf{U}})$$

2. Penyelesaian untuk perpindahan pada waktu $t + \Delta t$:

$$\mathbf{LDL}^T {}^{t+\Delta t}\mathbf{U} = {}^{t+\Delta t}\hat{\mathbf{R}}$$

3. Hitung perpindahan, kecepatan dan percepatan pada waktu $t + \Delta t$:

$$a. \quad {}^{t+\Delta t}\ddot{\mathbf{U}} = a_0({}^{t+\Delta t}\mathbf{U} - {}^t\mathbf{U}) - a_2 {}^t\dot{\mathbf{U}} - a_3 {}^t\ddot{\mathbf{U}}$$

$$b. \quad {}^{t+\Delta t}\dot{\mathbf{U}} = {}^t\dot{\mathbf{U}} + a_6 {}^t\ddot{\mathbf{U}} + a_7 {}^{t+\Delta t}\ddot{\mathbf{U}}$$

2.3 Matriks Penyusun Persamaan Analisis Dinamis

Persamaan 1 terbentuk dari tiga buah matriks sistem struktur, yaitu \mathbf{M} yaitu matriks massa, \mathbf{K} yaitu matriks kekakuan dan \mathbf{C} yaitu matriks redaman. Matriks tersebut tersusun dari *assembly* matriks elemen yang terbagi menjadi dua bagian yaitu yang pertama elemen struktur rangka dan kedua adalah elemen struktur bidang.

2.3.1 Matriks Kekakuan Elemen

Misalkan pada suatu elemen yang terbuat dari material homogen dan isotropik memiliki modulus elastisitas E dan rasio Poisson ν , kemudian luas penampang A dan momen inersia penampang terhadap sumbu yang melalui bidang netral dan sejajar sumbu z adalah I . Matriks kekakuan elemen ini adalah

$$\mathbf{k} = \begin{bmatrix} C_1 & 0 & 0 & -C_1 & 0 & 0 \\ 0 & 12C_2 & 6C_2L & 0 & -12C_2 & 6C_2L \\ 0 & 6C_2L & (4 + \varphi)C_2L^2 & 0 & -6C_2L & (2 - \varphi)C_2L^2 \\ -C_1 & 0 & 0 & C_1 & 0 & 0 \\ 0 & -12C_2 & -6C_2L & 0 & 12C_2 & -6C_2L \\ 0 & 6C_2L & (2 - \varphi)C_2L^2 & 0 & -6C_2L & (4 + \varphi)C_2L^2 \end{bmatrix} \quad (4)$$

$$C_1 = \frac{EA}{L} \quad \text{dan} \quad C_2 = \frac{EI}{L^3(1 + \varphi)}, \quad \varphi = \frac{12EI}{k_s AGL^2}, \quad \text{dan} \quad G = \frac{E}{2(1 + \nu)} \quad (5)$$

G pada persamaan 5 adalah modulus geser dan k_s adalah faktor koreksi geser yang bergantung pada bentuk penampang, contohnya untuk penampang berbentuk segiempat $k_s=5/6$.

Matriks kekakuan elemen rangka batang atau *truss element* juga dapat dilihat sebagai kasus khusus dari elemen rangka dengan $I=0$ dan semua rotasi pada titik *node*.

2.3.2 Matriks Massa Elemen

Misalkan elemen ini terbuat dari material yang homogen dan isotropik dengan massa jenis bahan ρ , rasio poisson ν dan luas penampang A , panjang elemen L dan momen inersia penampang terhadap sumbu yang melalui bidang netral dan sejajar sumbu z adalah I . Matriks massa elemen ini adalah

$$\mathbf{m} = \begin{bmatrix} 2t_1 & 0 & 0 & t_1 & 0 & 0 \\ 0 & 36t_2 & -(15\varphi-3)t_2L & 0 & -36t_2 & -(15\varphi-3)t_2L \\ 0 & -(15\varphi-3)t_2L & (10\varphi^2+5\varphi+4)t_2L^2 & 0 & (15\varphi-3)t_2L & (5\varphi^2-5\varphi-1)t_2L^2 \\ t_1 & 0 & 0 & 2t_1 & 0 & 0 \\ 0 & -36t_2 & (15\varphi-3)t_2L & 0 & 36t_2 & (15\varphi-3)t_2L \\ 0 & -(15\varphi-3)t_2L & (5\varphi^2-5\varphi-1)t_2L^2 & 0 & (15\varphi-3)t_2L & (10\varphi^2+5\varphi+4)t_2L^2 \end{bmatrix} + t_3 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (70\varphi^2+147\varphi+78) & (35\varphi^2+77\varphi+44)L/4 & 0 & (35\varphi^2+63\varphi+27) & -(35\varphi^2+63\varphi+26)L/4 \\ 0 & (35\varphi^2+77\varphi+44)L/4 & (7\varphi^2+14\varphi+8)L^2/4 & 0 & (35\varphi^2+63\varphi+26)L/4 & -(7\varphi^2+14\varphi+6)L^2/4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & (35\varphi^2+63\varphi+27) & (35\varphi^2+63\varphi+26)L/4 & 0 & (70\varphi^2+147\varphi+78) & -(35\varphi^2+77\varphi+44)L/4 \\ 0 & -(35\varphi^2+63\varphi+26)L/4 & -(7\varphi^2+14\varphi+6)L^2/4 & 0 & -(35\varphi^2+77\varphi+44)L/4 & (7\varphi^2+14\varphi+8)L^2/4 \end{bmatrix} \quad (6)$$

$$\text{Dengan } t_1 = \frac{\rho AL}{6}, t_2 = \frac{\rho I}{30(1+\nu)2L}, t_3 = \frac{\rho AI}{210(1+\nu)^2}, \varphi = \frac{12EI}{k_s AGL^2}, G = \frac{E}{2(1+\nu)} \quad (7)$$

Dimana G pada persamaan 7 adalah modulus geser dan k_s adalah faktor koreksi geser yang bergantung kepada bentuk penampang, contohnya untuk penampang berbentuk segiempat $k_s=5/6$.

2.3.3 Matriks Kekakuan Struktur dan Matriks Massa Struktur

Matriks kekakuan dan massa pada persamaan 4 dan persamaan 6, diturunkan dengan mengacu pada sistem koordinat lokal yang telah didefinisikan untuk setiap elemen. Untuk mendapatkan matriks kekakuan dan matriks massa elemen yang mengacu pada koordinat global dibutuhkan transformasi dari koordinat lokal ke global dengan persamaan sebagai berikut:

$$\mathbf{K} = \sum_{e=1}^{N_{elem}} (\mathbf{T}^e)^T \mathbf{k}^e \mathbf{T}^e \quad (8)$$

$$\mathbf{M} = \sum_{e=1}^{N_{elem}} (\mathbf{T}^e)^T \mathbf{m}^e \mathbf{T}^e \quad (9)$$

\mathbf{K} adalah matriks kekakuan struktur, \mathbf{M} adalah matriks massa struktur yang dinyatakan dalam koordinat global dan \mathbf{T} adalah matriks transformasi sebagai berikut:

$$\mathbf{T} = \begin{bmatrix} C & S & 0 & 0 & 0 & 0 \\ -S & C & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C & S & 0 \\ 0 & 0 & 0 & -S & C & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (10a)$$

$$C = \cos \theta, S = \sin \theta \quad (10b)$$

θ adalah sudut yang dibentuk antara sumbu x lokal dengan sumbu X global.

2.3.4 Keadaan Batas Dari Elemen Struktur

Penyelesaian persamaan analisis dinamis dapat diselesaikan dengan adanya pemisahan derajat kebebasan menjadi 2 kategori yaitu yang pertama derajat kebebasan aktif dan yang kedua derajat kebebasan pasif. Derajat kebebasan pasif yaitu derajat kebebasan yang sudah pasti, dimana perpindahan, kecepatan

dan percepatan akan ditentukan dari *input user*. Sebaliknya, bahwa derajat kebebasan aktif yaitu derajat kebebasan yang mengalami perpindahan, kecepatan dan percepatan yang diakibatkan karena gaya dari luar.

Kemudian setelah memilah kedua derajat kebebasan menjadi aktif dan pasif, akan digolongkan matriks \mathbf{K} dan matriks \mathbf{M} menjadi aktif dan pasif saja.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{22} & \mathbf{K}_{2p} \\ \mathbf{K}_{p2} & \mathbf{K}_{pp} \end{bmatrix}, \mathbf{M} = \begin{bmatrix} \mathbf{M}_{22} & \mathbf{M}_{2p} \\ \mathbf{M}_{p2} & \mathbf{M}_{pp} \end{bmatrix} \quad (11)$$

Berlaku juga dengan skalar perpindahan dengan percepatan akan menjadi:

$$\mathbf{U} = \begin{Bmatrix} \mathbf{u}_a \\ \mathbf{u}_p \end{Bmatrix}, \ddot{\mathbf{U}} = \begin{Bmatrix} \ddot{\mathbf{u}}_a \\ \ddot{\mathbf{u}}_p \end{Bmatrix} \quad (12)$$

Dengan mengabaikan redaman struktur terlebih dahulu, persamaan analisis dinamis berubah menjadi sebagai berikut:

$$\begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ap} \\ \mathbf{K}_{pa} & \mathbf{K}_{pp} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_a \\ \mathbf{u}_p \end{Bmatrix} + \begin{bmatrix} \mathbf{M}_{aa} & \mathbf{M}_{ap} \\ \mathbf{M}_{pa} & \mathbf{M}_{pp} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_a \\ \ddot{\mathbf{u}}_p \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_a \\ \mathbf{F}_p \end{Bmatrix} \quad (13)$$

Persamaan 13 di atas akan dipilah lagi sehingga persamaan menjadi seperti,

$$\mathbf{K}_{aa} \mathbf{U}_a + \mathbf{K}_{ap} \mathbf{U}_p + \mathbf{M}_{aa} \ddot{\mathbf{U}}_a + \mathbf{M}_{ap} \ddot{\mathbf{U}}_p = \mathbf{F}_a \quad (14)$$

Pada persamaan 14 diasumsikan bahwa seluruh perpindahan pasif yaitu nol (kondisi perpindahan pasif diasumsi hanya terjadi pada perletakan saja) sehingga persamaan menjadi sebagai berikut,

$$\mathbf{K}_{22} \mathbf{U}_2 + \mathbf{M}_{22} \ddot{\mathbf{U}}_2 = \mathbf{F}_2 \quad (15)$$

2.3.5 Matriks Redaman Struktur

Pada penelitian ini, rasio redaman struktur dianggap nol, sehingga matriks \mathbf{C} serta vektor kecepatan *input* dapat diabaikan, namun persamaan tetap menghasilkan *output* kecepatan yang juga berisi nol untuk setiap interval waktu.

2.4 HTML5

Suatu set kode revisi yang dibangun berdasarkan *HTML4* untuk menambahkan kemampuan/fitur yang baru. *HTML5* menawarkan banyak perbaikan dalam bidang penanganan aplikasi dan multimedia. Perbedaan terbesar adalah terdapat kode-kode baru untuk berbagai jenis konten yang sebelumnya telah ada tetapi dapat menggunakan kode yang lebih umum. Sebagai contoh, *HTML5* memiliki `<audio>` dan `<video>` tags untuk memasukkan konten multimedia, sedangkan *HTML4* memasukkan semua jenis multimedia konten melalui sebuah generik `<embed>` tag [6].

2.5 CSS3

CSS3 adalah versi terbaru dari *Cascading Style Sheets*. Bahasa *HTML* yang terlihat tidak menarik dapat menjadi menarik pada sebuah halaman *website* yang menggunakan *CSS3* mencakup seluruh versi sebelumnya yaitu *CSS2.1*, *CSS3* juga memiliki standar formal yang dapat didukung oleh semua *browser* yang digunakan secara luas [6].

2.6 PHP

PHP adalah bahasa pemrograman untuk membangun dinamis, situs *web* interaktif. Sebagai aturan umum, program *PHP* dijalankan pada *server web*, dan melayani halaman *web* untuk pengunjung berdasarkan permintaan. Salah satu fitur kunci *PHP* adalah bahwa dapat diimplementasikannya kode *PHP*

dalam halaman *web HTML*, sehingga sangat mudah untuk membuat konten yang dinamis dengan cepat [6].

PHP tersedia untuk berbagai sistem operasi dan *platforms*. *PHP* umumnya adalah halaman *HTML* dengan perintah *PHP* yang ada di dalamnya. *Web server* memproses perintah *PHP* dan mengirimkan *output* ke *browser*. *PHP* juga mendukung banyak sistem *database*, termasuk *MySQL*, *PostgreSQL*, *Oracle*, *Sybase*, *SQLite*, dan *ODBC-compliant databases* [6].

2.7 Framework CodeIgniter

CodeIgniter membantu menulis kode *PHP* dengan lebih baik, dan lebih mudah. Ini akan mengurangi jumlah kode yang perlu diketik. *Script* menjadi lebih mudah untuk dibaca dan di *update*. *CodeIgniter* akan membantu dalam pembuatan *website* besar yang memiliki struktur koheren. *CodeIgniter* juga akan membuat *coding* menjadi lebih sederhana.

CodeIgniter adalah sebuah *web application framework* yang bersifat *open source* digunakan untuk membangun aplikasi *php* dinamis. Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* dapat mengerjakan aplikasi lebih cepat daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah pengembangan aplikasi.

CodeIgniter sangat ringan, terstruktur, mudah dipelajari, dokumentasi lengkap dan dukungan yang luar biasa dari forum *CodeIgniter*. Selain itu *CodeIgniter* juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain:

- Menggunakan *Pattern MVC*
- *URL Friendly*
- Kemudahan.

MVC adalah konsep dasar yang harus diketahui sebelum mengenal *CodeIgniter*. *MVC* adalah singkatan dari *Model View Controller*. *MVC* sebenarnya adalah sebuah *pattern/teknik* pemrograman yang memisahkan *business logic* (alur pikir), *data logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana adalah memisahkan antara desain, data dan proses. Adapun komponen-komponen *MVC* antara lain:

1. Model

Model berhubungan dengan data dan interaksi ke *database* atau *webservice*. *Model* juga merepresentasikan struktur data dari aplikasi yang bisa berupa basis data maupun data lain, misalnya dalam bentuk *file* teks, *file XML* maupun *webservice*. Biasanya di dalam *model* akan berisi *class* dan fungsi untuk mengambil, melakukan *update* dan menghapus data *website*. Sebuah aplikasi *web* biasanya menggunakan basis data dalam menyimpan data, maka pada bagian *Model* biasanya akan berhubungan dengan perintah-perintah *query SQL*.

2. View

View berhubungan dengan segala sesuatu yang akan ditampilkan ke *end-user*. Bisa berupa halaman *web*, *rss*, *javascript* dan lain-lain. Harus dihindari adanya logika atau pemrosesan data di *view*. Di dalam *view* hanya berisi variabel-variabel yang berisi data yang siap ditampilkan. *View* dapat dikatakan sebagai halaman *website* yang dibuat dengan menggunakan *HTML* dan bantuan *CSS* atau *JavaScript*. Di dalam *view* jangan pernah ada kode untuk melakukan koneksi ke basis data. *View* hanya dikhususkan untuk menampilkan data-data hasil dari *model* dan *controller*.

3. Controller

Controller bertindak sebagai penghubung data dan *view*. Di dalam *controller* ini terdapat *class-class* dan fungsi-fungsi

yang memproses permintaan dari *View* ke dalam struktur data di dalam *Model*. *Controller* juga tidak boleh berisi kode untuk mengakses basis data karena tugas mengakses data telah diserahkan kepada *model*. Tugas *controller* adalah menyediakan berbagai variabel yang akan ditampilkan di *view*, memanggil *model* untuk melakukan akses ke basis data, menyediakan penanganan kesalahan/*error*, mengerjakan proses logika dari aplikasi serta melakukan validasi atau cek terhadap *input* [6].

2.8 JavaScript

JavaScript digunakan untuk memperkaya tampilan pada *HTML*, yang merupakan skrip program berbasis *client* yang dieksekusi oleh *browser* sehingga membuat halaman *web* dapat melakukan tugas-tugas tambahan yang tidak bisa dilakukan oleh skrip *HTML* biasa.

Kode *javascript* dituliskan dalam bentuk fungsi yang ditaruh di *tag* `<head>` yang dibuka dengan *tag* `<script type="text/javascript">`. Kode *javascript* juga dapat diletakkan di *file* tersendiri yang berekstensi *.js* (singkatan dari *javascript*). Untuk memanggil kode *javascript* yang terdapat di *file* sendiri, di bagian awal `<head>` harus ditemukan dahulu nama *file .js*.

2.9 MySQL

MySQL adalah pembuatan *database* yang memungkinkan *PHP* dan *Apache* untuk bekerja bersama-sama untuk mengakses dan menampilkan data dalam format yang dapat dibaca ke *browser*. *MySQL* adalah *Structured Query Language server* yang dirancang untuk pengolahan *query* yang kompleks. Sebagai sistem *database* relasional, *MySQL* memungkinkan banyak tabel yang berbeda untuk bergabung bersama-sama untuk meningkatkan efisiensi maksimum dan kecepatan.

MySQL memiliki kecepatan dan ukuran kecil sehingga ideal untuk situs web. *MySQL* memiliki beberapa keuntungan seperti:

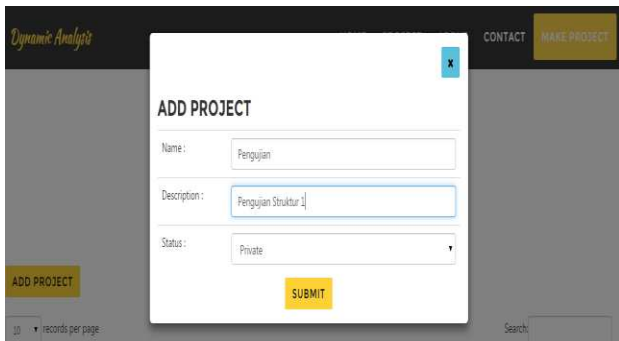
- Cepat
- Murah
- Mudah untuk digunakan.
- Dapat digunakan pada banyak sistem operasi.
- *Secure*.
- Mendukung *database* yang besar.

2.10 Paper.js

Paper.js adalah *framework* skrip grafis vektor *open source* yang berjalan di atas *canvas HTML5*. *Paper.js* menawarkan *scene graph / Document Object Model* dan banyak fungsi yang lain untuk membuat grafis vektor dan kurva, serta membuat *user interface* program menjadi lebih baik, rapi dan konsisten.

Paper.js menawarkan :

- Sebuah *scene graph / document object model* untuk grafis vektor: bekerja dengan *nested layers*, *groups*, *paths*, *compound paths*, *rasters*, *symbol*, dll.
- Penanganan dan penggambaran dari *item* grafis yang otomatis dan dioptimalkan, memungkinkan untuk membangun atau memodifikasi *item* dan *styles* serta menyerahkan *drawing commands* kepada *paper.js*.
- *Application Programming Interface (API)* yang dirancang dengan baik.
- *PaperScript*, sebuah ekstensi sederhana dari *JavaScript*, yang mengizinkan lingkup eksekusi untuk *script* tanpa mengganggu lingkup global, eksekusi untuk beberapa skrip per halaman di dalam lingkup *sand-boxed* yang terpisah sambil membagikan *library code*, dan menambahkan dukungan untuk operator *overloading* kepada objek apapun.



Gambar 3. Hasil Pembuatan Project

4.1.2 Tab Input Node

Proses *input* perhitungan dapat dimulai dari *input node* untuk menentukan titik koordinat pada setiap *node* yang akan dibuat dan tipe *boundary condition* dari *node* tersebut. Pengujian dilakukan dengan node 1 pada x coordinate 0, y coordinate 0, *boundary condition fixed*, node 2 pada x coordinate 0, y coordinate 180, *boundary condition Roll 1* dan node 3 pada x coordinate 0, y coordinate 300, *boundary condition Roll 1*. Hasil *input node* pada tabel dapat dilihat pada Gambar 4.

Nodes	Properties	Element	Joint Mass	Joint Loads	Ground Acceleration	Proses
Node						
New Destroy Save Cancel						
	X Coordinate	Y Coordinate	Boundary Condition			
1	0	0	Fixed			
2	0	180	Roll 1			
3	0	300	Roll 1			

Gambar 4. Hasil Input Node Pada Tabel

4.1.3 Tab Input Element

Proses *input element* dilakukan dengan memilih *node* pertama dan *node kedua* yang akan dihubungkan menjadi satu *element*. Pengujian dilakukan dengan element 1 *first node* pada *node 1* dan *second node* pada *node 2* dengan *property property 1* dan element 2 *first node* pada *node 2* dan *second node* pada *node 3* dengan *property property 2*. Hasil *input elements* pada tabel dapat dilihat pada Gambar 5.

Nodes	Properties	Element	Joint Mass	Joint Loads	Ground Acceleration	Proses
Elements						
New Destroy Save Cancel						
	First Node	Second Node	Property			
1	Node #1	Node #2	Property #1			
2	Node #2	Node #3	Property #2			

Gambar 5. Hasil Input Element Pada Tabel

4.1.4 Tab Input Joint Loads

Joint loads merupakan beban yang diberikan tiap *interval* waktu pada setiap *degree of freedom aktif* pada setiap *node*. Pengujian dilakukan dengan target *node* pada *node 2*, hasil tabel *input joint loads* dapat dilihat pada Gambar 6.

Nodes	Properties	Element	Joint Mass	Joint Loads	Ground Acceleration	Proses
Joint Loads						
New Destroy Save Cancel						
	Target Node	Fx	Fy	M		
1	Node #1	no	no	no		
2	Node #2	yes	no	no		
3	Node #3	yes	no	no		

FX NODE 2
FX NODE 3

Gambar 6. Hasil Input Joint Loads Pada Tabel

4.1.5 Tab Input Ground Acceleration

Pengujian pada *input ground acceleration* dilakukan dengan t1-t11 dengan nilai 0 hingga 0.1, a vertikal pada t1-t11 dengan nilai 0 dan a horizontal pada t1-t11 dengan nilai 10000000. Hasil *input ground acceleration* dapat dilihat pada Gambar 7.

Nodes	Properties	Element	Joint Mass	Joint Loads	Ground Acceleration	Proses
Ground Acceleration						
New Destroy Save Cancel						
	t	a vertikal	a horizontal			
1	0	0	10000000			
2	0.01	0	10000000			
3	0.02	0	10000000			
4	0.03	0	10000000			
5	0.04	0	10000000			
6	0.05	0	10000000			
7	0.06	0	10000000			
8	0.07	0	10000000			
9	0.08	0	10000000			
10	0.09	0	10000000			
11	0.1	0	10000000			

Gambar 7. Hasil Input Ground Acceleration

4.1.6 Hasil Calculate Solution Phase

Hasil *Solution phase* pada pengujian untuk kecepatan dapat dilihat pada Gambar 8.

+ Solution Phase												
+ Vektor U												
t	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1	
d2x	0	0.01881	0.07374	0.16041	0.27275	0.40438	0.54871	0.69921	0.84951	0.99357	1.12581	1.24117
Q2	0	-0.0001	-0.0004	-0.0009	-0.0016	-0.0025	-0.00359	-0.00485	-0.00625	-0.00774	-0.00926	-0.01076
d3x	0	0.03496	0.13749	0.30051	0.51447	0.76947	1.0555	1.36258	1.68089	2.00084	2.31308	2.60854
Q3	0	-0.00014	-0.00057	-0.00127	-0.00221	-0.00338	-0.00473	-0.0062	-0.00776	-0.00932	-0.01084	-0.01226

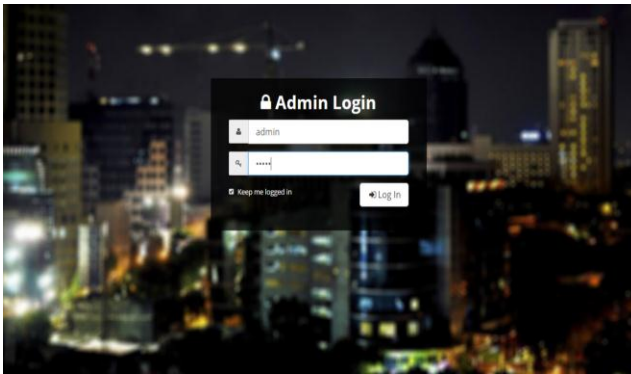
Gambar 8. Hasil Kecepatan Pada Solution Phase

4.2 Pengujian Sistem Back-End Website untuk Admin

4.2.1 Halaman Login Admin

Admin website dapat mengakses halaman *dashboard* dengan melakukan proses *login* terlebih dahulu, pengujian *login admin*

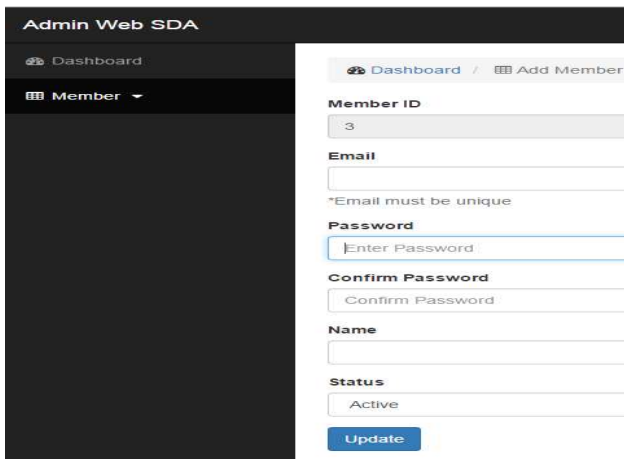
dilakukan dengan *username admin* dan *password admin* dapat dilihat pada Gambar 9.



Gambar 9. Halaman Login Admin

4.2.2 Halaman Dashboard Admin Member

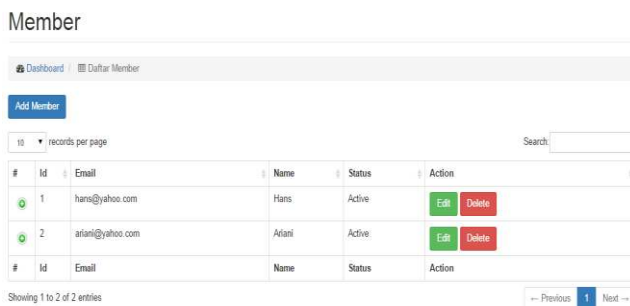
Admin dapat melakukan pendaftaran akun baru untuk *user website*, pengujian dilakukan dengan *e-mail hans@yahoo.com*, *password admin*, *confirm password admin*, *name hans*, *status active* seperti pada Gambar 10.



Gambar 10. Halaman Dashboard Add Member

4.2.3 Halaman Dashboard Daftar Member

Pengujian yang dilakukan pada penambahan akun *user* baru yang berhasil, maka akan terlihat seperti pada Gambar 11.



Gambar 11. Halaman Dashboard Daftar Member

5. KESIMPULAN & SARAN

5.1 Kesimpulan

Dari hasil perancangan dan pembuatan *website Dynamic Analysis of 2D Frame* dapat diambil beberapa kesimpulan sebagai berikut :

- Dengan menggunakan *library JAMA* untuk membantu menyelesaikan operasi matriks pada algoritma kedalam bahasa pemrograman pada pembuatan *website Dynamic Analysis of 2D Frame*.
- Dengan implementasikan metode *Newmark* yang telah diuji secara teori dan praktek dapat menyelesaikan persamaan analisis dinamis yang menghasilkan *output* data berupa perpindahan serta deformasi yang dialami oleh struktur akibat gaya dinamis.
- *Website* analisis dinamis struktur rangka dua dimensi untuk bangunan tinggi dengan hasil perhitungan yang akurat dapat dibuat dengan mengembangkan *website* ta29 yang merupakan analisis statis struktur rangka dua dimensi. Pengembangan yang dilakukan antara lain seperti pengurangan dan penambahan *input*, serta perubahan simulasi dengan *interval* waktu.
- Verifikasi untuk *website Dynamic Analysis of 2D Frame* tidak bisa menggunakan bantuan *SAP2000* karena perbedaan metode yang digunakan dalam menyelesaikan analisis dinamis (*Newmark* versus *Hughes, Hybler, Tyler's method*) serta perbedaan untuk proses *input* dan hasil *output*.

5.2 Saran

Saran yang dapat diberikan untuk pengembangan program aplikasi perancangan dan pembuatan *website Dynamic Analysis of 2D Frame* adalah :

- Mengimplementasikan redaman struktur.
- Mengimplementasikan perpindahan, kecepatan, percepatan dan gaya dengan ketergantungan waktu pada derajat kebebasan pasif.

6. REFERENSI

- [1] Bathe, K.J. (1996). *Finite Element Procedures*. America: Prentice-Hall, Inc.
- [2] Doyle, M. 2010. *Beginning PHP 5.3*. Indianapolis: Wiley Publishing, Inc.
- [3] Id, I. D. 2011. *Framework Codeigniter Sebuah Panduan dan Best Practice*. URI = http://www.academia.edu/3647990/Framework_Codeigniter.pdf
- [4] Powers, D. 2012. *Beginning CSS3*. New York: Apress
- [5] Tatroe, K., MacIntyre, P. & Lerdorf, R. 2013. *Programming PHP, 3rd Edition*. Sebastopol: O'Reilly Media, Inc.
- [6] Wempen, F. 2011. *HTML5 Step by Step*. Sebastopol: O'Reilly Media, Inc.