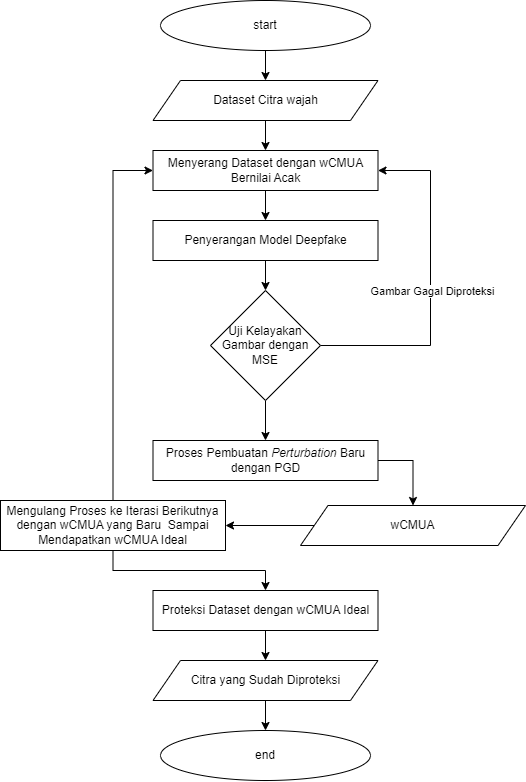
### Kerangka Kerja

Kerangka kerja pada penelitian ini adalah sebagai berikut:

  
Gambar III-2. Diagram Alir Sistem Proteksi Citra dari deepfake dengan metode CMUA

Berdasarkan kerangka kerja penelitian pada Gambar III-2, sistem proteksi dari *deepfake* dengan metode CMUA memiliki alur sistem sebagai berikut:

**Input Dataset Citra Wajah**

Proses penginputan dataset yang digunakan dalam sistem ini, yaitu berupa citra gambar wajah dari dataset Celeb-A yang digunakan sebagai data latih.

**Menyerang Dataset dengan wCMUA Bernilai Acak**

Memasangkan proteksi awal dengan perturbation awal yang bernilai acak

**Penyerangan Model *Deepfake***

Melakukan penyerangan terhadap dataset yang belum diproteksi dan dataset yang yang telah di proteksi

***Mean Square Error* (MSE)**

Pada tahapan ini gambar yang sebelumnya sudah diserang oleh model deepfake akan di uji dengan metode MSE untuk mengukur perbedaan antara *G*(*I1*)... *G*(*In*)(gambar yang deepfake tanpa proteksi) dan *G*(*I1* + *W*)... *G*(*In* + *W*)(gambar yang telah diproteksi terlebih dahulu sebelum di deepfake), di mana E adalah nilai batas atas dari watermark lawan W.

**PGD**

menggunakan PGD (Madry et al. 2018) sebagai metode serangan dasar untuk memperbarui *adversarial perturbations* pada setiap iterasi serangan,

**Adversarial Perturbation Fusion**

Konflik di antara watermark yang berlawanan yang dihasilkan dari gambar dan model yang berbeda akan mengurangi kemampuan transferabilitas *CMUA-Watermark* yang diusulkan. Untuk melemahkan konflik ini, digunakan strategi fusi gangguan dua tingkat selama proses serangan. Secara khusus, ketika menyerang satu model *deepfake* tertentu, akan melakukan **fusi tingkat gambar** untuk merata-rata gradien yang di *sign* dari sekumpulan gambar wajah,

di mana *bs* adalah ukuran kumpulan gambar wajah, dan *Iadvj* adalah *adversarial image* ke-j dari sebuah *batch*. Operasi ini akan menyebabkan *Gavg* lebih berkonsentrasi pada atribut umum wajah manusia daripada atribut wajah tertentu. Kemudian, menggunakan PGD untuk menghasilkan adversarial perturbation *Pavg* melalui *Gavg*.

Setelah mendapatkan *Pavg* dari satu model, melakukan **fusi tingkat model**, yang secara iteratif menggabungkan *Pavg* yang dihasilkan dari model tertentu ke *WCMUA* dalam pelatihan, dan *WCMUA* awal hanyalah *Pavg* yang dihitung dari model deepfake pertama,

,

di mana *α* adalah faktor peluruhan, *Ptavg* adalah rata-rata gangguan yang dihasilkan dari model *deepfake* yang diserang ke-*t*, dan *Wt*CMUA adalah *CMUA-Watermark* pelatihan setelah model *deepfake* yang diserang ke-*t*.

**Automatic Step Size Tuning based on TPE**

Selain fusi dua tingkat yang disebutkan di atas, ditemukan bahwa ukuran langkah serangan untuk model yang berbeda juga penting untuk transferabilitas CMUA-Watermark yang dihasilkan. Oleh karena itu, mengeksploitasi pendekatan heuristik untuk secara otomatis menemukan ukuran langkah serangan yang sesuai.

Metode serangan dasar yang dipilih (PGD) termasuk ke dalam keluarga FGSM (Goodfellow, Shlens, dan Szegedy 2015), dan gradien *∇xL* dinormalisasi oleh fungsi *sign*:

Dalam perhitungan nyata, elemen-elemen dalam *∇xL* hampir tidak pernah mencapai 0, sehingga ||sign(*∇xL*)||2 ≈ 1 adalah tetap untuk setiap gradien. Perturbasi *∆P* yang diperbarui dalam iterasi metode serangan berbasis *sign* dirumuskan sebagai:

Dengan kata lain, hanya ukuran langkah *a* yang menentukan tingkat pembaruan selama serangan, sehingga pemilihan *a* memiliki pengaruh yang besar terhadap performa serangan. Kesimpulan ini juga berlaku untuk serangan universal lintas model; perturbasi yang diperbarui *∆Pu* dalam sebuah iterasi serangan universal lintas model dibentuk dengan menggabungkan *∆Pi* dari beberapa model *G1*,..., *Gm*:

Dalam rumus di atas, *m* adalah jumlah model, faktor peluruhan α adalah sebuah konstanta, dan *sign* (*∇XLi*) memberikan arah optimasi untuk *Gi* . Oleh karena itu, arah optimasi secara keseluruhan sangat dipengaruhi oleh *a1*, ..., *am*, dan memilih *a1*, ..., *am* yang sesuai di berbagai model untuk menemukan arah keseluruhan yang ideal adalah masalah utama untuk serangan lintas model.

Menggunakan algoritma TPE (Bergstra et al. 2011) untuk memecahkan masalah ini, yang secara otomatis mencari *a1*, ..., *am* yang sesuai untuk menyeimbangkan arah yang berbeda yang dihitung dari berbagai model. TPE adalah metode optimasi hiper-parameter berdasarkan *Sequential Model-Based Optimization* (SMBO), yang secara berurutan membangun model untuk memperkirakan kinerja hiper-parameter berdasarkan pengukuran historis, dan kemudian memilih hiperparameter baru untuk diuji berdasarkan model ini. Dalam penelitian ini, menganggap ukuran langkah *a1*, ..., *am* sebagai hyperparameter input *x* dan tingkat keberhasilan serangan sebagai nilai kualitas y dari TPE. TPE menggunakan *P*(*x*|*y*) dan *P*(*y*) untuk memodelkan *P*(*y*|*x*), dan *p*(*x*|*y*) diberikan oleh:

di mana *y*∗ ditentukan oleh pengamatan terbaik secara historis, e(*x*) adalah densitas yang dibentuk dengan pengamatan {*x*(*i*)} sedemikian rupa sehingga kerugian yang sesuai lebih rendah dari *y*∗ , dan *g*(*x*) adalah densitas yang dibentuk dengan pengamatan yang tersisa. Setelah memodelkan *P*(*y*|*x*), lalu terus mencari ukuran langkah yang lebih baik dengan mengoptimalkan kriteria *Expected Improvement* (EI) di setiap iterasi pencarian, yang diberikan oleh,

di mana γ = p ( y < y\* ). Dibandingkan dengan kriteria lainnya, EI bersifat intuitif dan telah terbukti memiliki kinerja yang sangat baik. Untuk detail lebih lanjut mengenai TPE, lihat (Bergstra et al. 2011).